Navy Personnel Research and Development Center

San Diego, CA 92152-6800 TN-90-30 July 1990

2

DTIC FILE COPY

AD-A225 706

Computer-based Instruction Authoring Tools System (CATS): Lesson Maintenance

Vel N. Hulton
Michael R. Flaningam
Barbara Tarker
Ann Rybowiak
Susan Sulzbach
Mark Lyon
Brian Thomason

DTIC
ELECTE
AUG 24 1990
S E D

Approved for public release; distribution is unlimited.

90 08 23 041

NPRDC-TN-90-30 July 1990

Computer-based Instruction Authoring Tools System (CATS): Lesson Maintenance

Vel N. Hulton
Michael R. Flaningam
Barbara Tarker
Ann Rybowiak
Navy Personnel Research and Development Center

and

Susan Sulzbach
Mark Lyon
Brian Thomason
Systems Engineering Associates, San Diego

Reviewed and released by
Steven L. Dockstader
Director, Organizational Systems Department

Accession For
NTIS GRA&I ☒
DTIC TAB □
Unannounced □
Justification
By
Distribution/
Availability Codes
Dist | Avail and/or Special
A-1

Approved for public release;
distribution is unlimited.

Navy Personnel Research and Development Center
San Diego, California 92152-6800

QUALITY INSPECTED 4

REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 0704-0188

Public reporting burden for this collection of information is estimated to average 1 hour per response, including the time for reviewing instructions, searching existing data sources, gathering and maintaining the data needed, and completing and reviewing the collection of information. Send comments regarding this burden estimate or any other aspect of this collection of information, including suggestions for reducing this burden, to Washington Headquarters Services, Directorate for Information Operations and Reports, 1215 Jefferson Davis Highway, Suite 1204, Arlington, VA 22202-4302, and to the Office of Management and Budget, Paperwork Reduction Project (0704-0188), Washington, DC 20503.

1. AGENCY USE ONLY *(Leave blank)*

2. REPORT DATE
July 1990

3. REPORT TYPE AND DATE COVERED
Tech. Note--Jan 1989-Jun 1990

4. TITLE AND SUBTITLE
Computer-based Instruction Authoring Tools System (CATS):
Lesson Maintenance

5. FUNDING NUMBERS
Program Element 0604722A

6. AUTHOR(S)
V. N. Hulton, M. R. Flaningam, B. Tarker, A. Rybowiak, S. Sulzbach, M. Lyon, & B. Thomason

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)
Navy Personnel Research and Development Center
San Diego, California 92152-6800

8. PERFORMING ORGANIZATION REPORT NUMBER
NPRDC-TN-90-30

9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)
Deputy Chief of Naval Operations (MPT)
Assistant for Planning and Technology Development (OP-01B2)
Washington, DC 20301

10. SPONSORING/MONITORING AGENCY REPORT NUMBER

11. SUPPLEMENTARY NOTES
See also NPRDC TN-90-27, TN-90-28, AND TN-90-29.

12a. DISTRIBUTION/AVAILABILITY STATEMENT
Approved for public release; distribution is unlimited.

12b. DISTRIBUTION CODE

13. ABSTRACT *(Maximum 200 words)*

This publication is one of three technical notes that supplement the CATS (Computer-Based Instruction Authoring Tools System) user's manual. This document describes the lesson maintenance system and details procedures for its use. Intended readers are CATS authors and instructional developers who update CATS courseware.

14. SUBJECT TERMS
Computer-based instruction, CBI, computer-based instruction authoring tools system, CATS, transportable education, authoring system, courseware maintenance, interactive software

15. NUMBER OF PAGES
68

16. PRICE CODE

17. SECURITY CLASSIFICATION OF REPORT
UNCLASSIFIED

18. SECURITY CLASSIFICATION OF THIS PAGE
UNCLASSIFIED

19. SECURITY CLASSIFICATION OF ABSTRACT
UNCLASSIFIED

20. LIMITATION OF ABSTRACT
UNLIMITED

FOREWORD

This technical note supplements the Computer-Based Instruction Authoring Tools System (CATS) user's manual (NPRDC Tech. Note 90-27). It is intended for instructional developers who update CATS courseware.

CATS is a government-owned set of software tools for constructing and presenting interactive courseware on a personal computer. It was developed by Systems Engineering Associates (SEA), San Diego, through a series of contracts with the Navy Personnel Research and Development Center (NAVPERSRANDCEN). CATS has continued to evolve because of the relative ease with which a government-owned system can be modified and enhanced to meet the needs of new government projects.

This publication is one of three technical notes that supplement the user's manual. This document describes the lesson maintenance system. A second report addresses the presentation system using a prototype software program called Lesson Manager (NPRDC Tech. Note 90-28). A third report is intended for students who are taking the computer-based version of the Defense Systems Management College's Funds Management and Tools for Program Management lessons. The objective of this effort was to provide instructions for the use of the disks that the students are given to run each lesson (NPRDC Tech. Note 90-29).

These three reports were prepared by SEA and NAVPERSRANDCEN as part of the Guidelines for Transportable Education and Training (GTET) project. The Joint Services Manpower and Training Systems Development Program funded the project through Program Element 0604722A. The primary objective of GTET is to develop guidelines for transportable education and training that can be used by those who wish to develop and implement transportable instructional packages. These supplemental reports support that objective.

Questions regarding this work can be directed to Dr. Michael R. Flaningam, Code 162, Organizational Systems Department, Navy Personnel Research and Development Center, (619) 553-0554 or AUTOVON 553-0554. Requests for system software should be addressed to Dr. Kirk Johnson, Code 14, Navy Personnel Research and Development Center, (619) 553-7702 or AUTOVON 553-7702.

STEVEN L. DOCKSTADER
Director, Organizational Systems Department

SUMMARY

Background

The Computer-Based Instruction Authoring Tools System (CATS) was developed under contract for the Navy Personnel Research and Development Center. The objective of this effort was to create and test low-cost computer-based instructional systems. CATS proved to be the means to that end. CATS has continued to evolve because of the relative ease with which a government-owned system can be modified and enhanced to meet the emerging needs of new government work.

System Capabilities

CATS materials can be delivered on Zenith 100 series microcomputers and IBM PC/XT/AT-compatible microcomputers operating under MS-DOS (Microsoft disk operation[1] system). Materials can be authored on these same machines as well as on VAX and Sun computers operating under UNIX. Materials can be authored and compiled on one type of machine and delivered on another type. There are separate versions of the delivery software for machines with Z-100, CGA, EGA, Tecmar, and Visage graphics boards.

The system provides drivers for a Microsoft compatible mouse, a MicroTouch touch screen, and an Hitachi VIP9500 videodisc player (all are optional). Additional drivers can be developed and added to the system at minimal cost.

The courseware is controlled by a series of storyboards. The storyboard is the building block of the CATS system. It is a block of text that specifies what the computer should display in a particular screen image, how it should react to student responses to that screen image, and anything else it should do while the image is displayed. Storyboards can be written with any word processing program that can save files in an unformatted ASCII text mode. Still graphics can be developed with any graphics program, provided the graphic can be captured by Dr. Halo, a graphics software program for IBM-compatible computers. Animation can be provided by several commercial animation software packages.

CATS software does two things. First, it compiles text files, reducing them to a more compact form. This is done before the lesson is presented to the student. Second, it interprets the compiled files, creating the commands that actually control the computer while the lesson is being presented to the student.

There are no prompts or menus to aid the author during lesson development. In fact, CATS is not active while the storyboards are being written. Repetitive features of a lesson, ranging from parts of a storyboard to a series of complete storyboards, can be readily copied from one part of the lesson to another or from special templates created by the author.

[1] Identification of specific equipment and software is for documentation only and does not imply endorsement.

Users

The system can be used without fee by any activity of the Federal Government or by any contractor who is developing instructional materials for exclusive use by the Federal Government.

Documentation

Three technical notes have been written to supplement the CATS user's manual (NPRDC Tech. Note 90-27). This report focuses on maintenance of CATS courseware. A second report concerns the presentation system and the use of a prototype software program called Lesson Manager (NPRDC Tech. Note 90-28). The third supplemental report is a guide for students who are taking the computer-based version of the Defense Systems Management College's Funds Management and Tools for Program Management lessons (NPRDC Tech. Note 90-29). The objective of this effort was to provide instructions for the use of the disks that the students are given to run each lesson.

Future Efforts

The user's manual will continue to be revised as changes are made to CATS. Users are urged to contact either the Defense Technical Information Center, Building #5, Cameron Station, Alexandria, Virginia 22304-6145, or Code 14, Navy Personnel Research and Development Center, about updated versions.

CONTENTS

Page

Page

LIST OF TABLES

Page

LIST OF FIGURES

INTRODUCTION

WHAT IS THIS MANUAL ABOUT?

This document describes the maintenance of computer-based instruction developed by the Navy Personnel Research and Development Center (NPRDC) using the Computer-based Instruction Authoring Tools System (CATS).

Presentation and maintenance of CATS courseware are controlled by two different systems.

1. The **lesson presentation system** allows CATS courseware to be **presented** to students. CATS lessons can be installed and presented by using a lesson management software program called Lesson Manager. Procedures for using Lesson Manager are described in *Computer-Based Instruction Authoring Tools System (CATS): Lesson Presentation* (NPRDC Tech. Note 90-28). CATS lessons can also be installed and presented without using this special software. Procedures for installing and presenting CATS lessons without Lesson Manager are given in *Computer-Based Instruction Authoring Tools System (CATS): Student Guide* (NPRDC Tech. Note 90-29).

2. The **lesson maintenance system** allows CATS instructional developers to **create** presentation disks and to **update** CATS courseware. The lesson maintenance system is a central repository for all CATS courseware. This document describes the **lesson maintenance system** and details procedures for its use.

WHO WILL USE THIS MANUAL?

This publication supplements the CATS user's manual (NPRDC TN-90-27) and is intended to be used by instructional developers who are experienced computer users and CATS authors. These personnel should already understand:

- How to use common MS-DOS commands (i.e., copy, rm)
- The difference between MS-DOS files and directories
- How to create and name MS-DOS files
- How to use, create, and change MS-DOS tree-structured directories
- How to use, create, and change MS-DOS batch files
- How to use floppy disks
- How to use a full-screen text editor or word processor
- How to create CATS storyboards
- How to use `CATSC`, `CATSM`, and `CATSVIEW`
- What a CATS profile file is

To help the reader in understanding the concepts and terminology associated with CATS, a glossary is included at the end of the report.

LESSON MAINTENANCE SYSTEM COMPONENTS

The lesson maintenance system consists of the following components:

1. **Hardware:**

   Zenith 100 series and IBM PC/XT/AT-compatible microcomputers
   Low-density floppy disk drive

2. **System Software:**

   MS-DOS
   CATS system files
   Word processor
   Graphics editor
   Animation editor

3. **CATS Courseware.** The files used to create and present computer-based instruction (CBI) with the CATS:

   Storyboard files
   Graphic files
   Animation files
   CATS variable files
   Batch files
   Segment files
   Profile files
   DIR files

This document addresses details about the system's coursewaie only. The CATS user's manual offers detailed information about the system's hardware and software.

CATS COURSEWARE FILE ORGANIZATION

During the early stages of courseware development, the courseware developer creates an instructional outline for the courseware material. Figure 1 illustrates a partial outline dividing sample instructional material into three lessons on the subject of word processing to be presented as CBI using the CATS. Each lesson is divided into topics, and each topic into drill, practice, and test components. Each drill is divided into questions.

1.CBI Training: *Word Processing*

1.1 Lesson: *Getting Started*

1.1.a Topic: *Running the Program*

1.1.a.1 Drill

1.1.a.1.1 Question 1

1.1.a.1.2 Question 2

⋮

1.1.a.1.10 Question 10

1.1.a.2 Practice

⋮

1.1.a.3 Test

⋮

1.1.b Topic: *Creating a File*

⋮

1.1.c Topic: *Saving a File*

⋮

1.2. Lesson: *Creating Documents*

⋮

1.3. Lesson: *Advanced Techniques*

⋮

Figure 1. CATS courseware is organized using outlines.

CBI requires a very detailed outline of its instructional material.

CATS courseware files are stored on the system's hard disk. Hundreds of courseware files may be required to create and present moderate-to-large CATS lessons. These files are organized on the hard disk using an MS-DOS directory tree whose structure is based on the courseware's instructional outline.

Figure 2 shows a diagram of the MS-DOS directory tree for the material in Figure 1. Each box represents a directory. Each box contains the directory name and the item's name found in the outline (Figure 1).

The first row in Figure 2 contains a single directory that corresponds to the first level in the outline in Figure 1. The second row of directories corresponds to the second level of items (lessons) in the outline. The third row of directories corresponds to the third level of items (topics) in the outline. The lowest level of items in the outline are files. The names attached to the directory called DRILL are files that correspond to the most subordinate items of lesson 1 in the outline (i.e., the drill questions). (Owing to DOS limitations on file name length, the names of directories and files may have to be abbreviated forms of the outline names.)

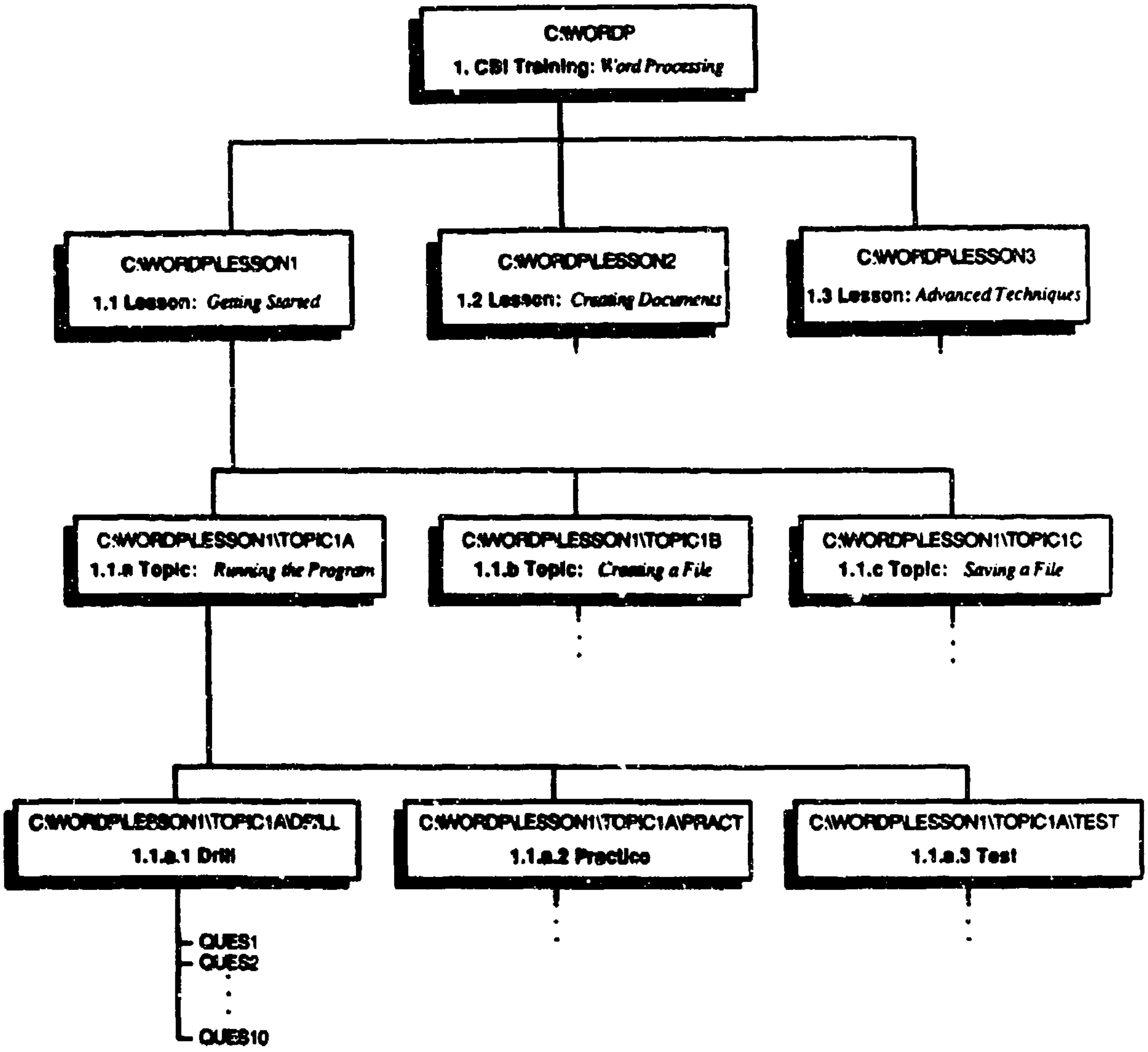

C:\WORDP
1. CBI Training: *Word Processing*
C:\WORDP\LESSON1
1.1 Lesson: *Getting Started*
C:\WORDP\LESSON2
1.2 Lesson: *Creating Documents*
C:\WORDP\LESSON3
1.3 Lesson: *Advanced Techniques*
C:\WORDP\LESSON1\TOPIC1A
1.1.a Topic: *Running the Program*
C:\WORDP\LESSON1\TOPIC1B
1.1.b Topic: *Creating a File*
C:\WORDP\LESSON1\TOPIC1C
1.1.c Topic: *Saving a File*
C:\WORDP\LESSON1\TOPIC1A\DRILL
1.1.a.1 Drill
C:\WORDP\LESSON1\TOPIC1A\PRACT
1.1.a.2 Practice
C:\WORDP\LESSON1\TOPIC1A\TEST
1.1.a.3 Test
QUES1
QUES2
QUES10

Figure 2. MS-DOS directory trees organize CATS courseware files.

Figure 3 shows another outline and its corresponding directory tree. This figure suggests that items in an outline do not need to be called lessons, topics, etc. They can be called anything, as long as it's done in a consistent, systematic way. For example, items in the second level are called courses, whereas items in the second level of Figure 1 are called lessons.

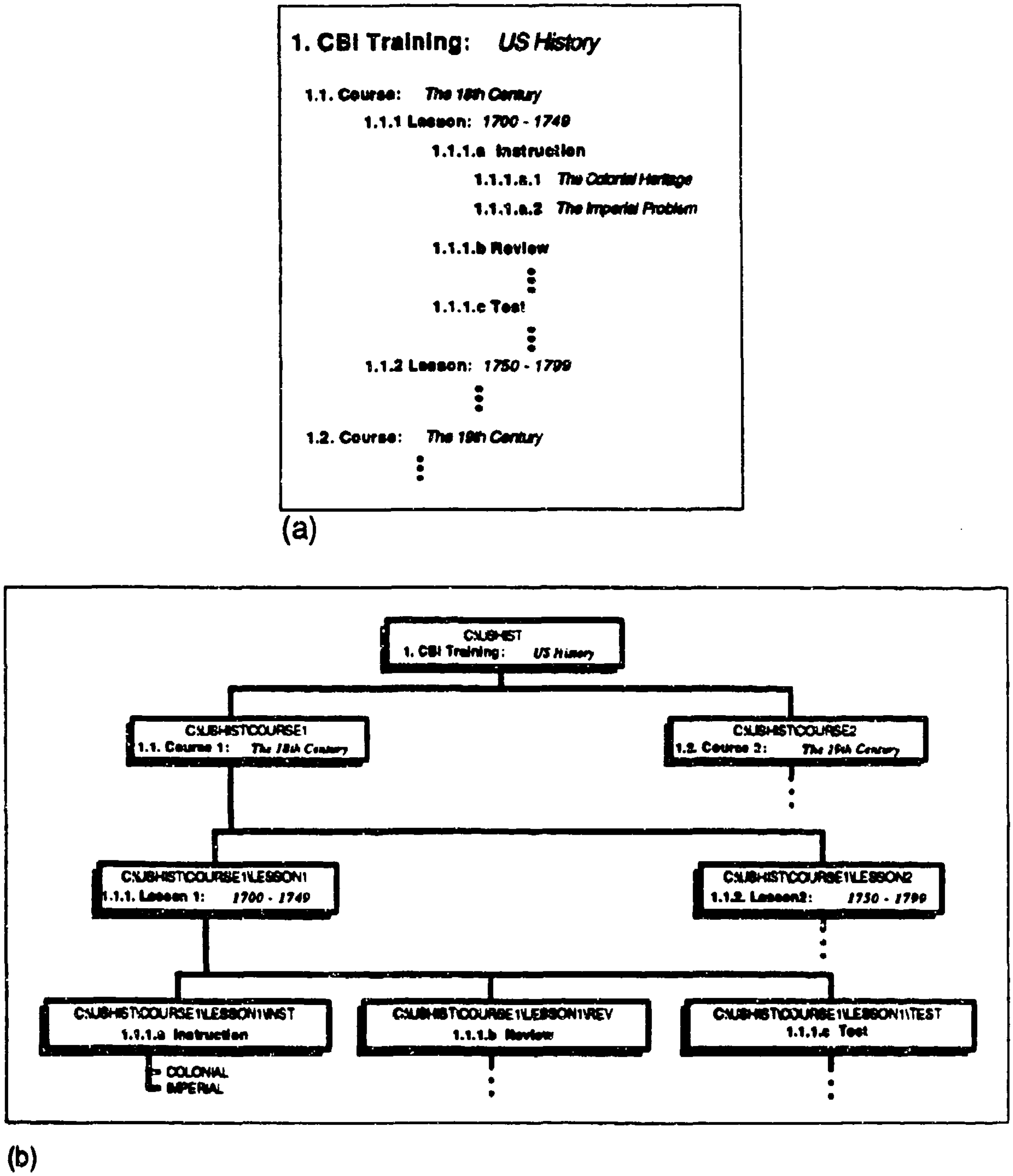

1. CBI Training: *US History*
1.1. Course: *The 18th Century*
1.1.1 Lesson: *1700 - 1749*
1.1.1.a Instruction
1.1.1.a.1 *The Colonial Heritage*
1.1.1.a.2 *The Imperial Problem*
1.1.1.b Review
1.1.1.c Test
1.1.2 Lesson: *1750 - 1799*
1.2. Course: *The 19th Century*
(a)
C:\USHIST
1. CBI Training: *US History*
C:\USHIST\COURSE1
1.1. Course 1: *The 18th Century*
C:\USHIST\COURSE2
1.2. Course 2: *The 19th Century*
C:\USHIST\COURSE1\LESSON1
1.1.1. Lesson 1: *1700 - 1749*
C:\USHIST\COURSE1\LESSON2
1.1.2. Lesson2: *1750 - 1799*
C:\USHIST\COURSE1\LESSON1\INST
1.1.1.a Instruction
COLONIAL
IMPERIAL
C:\USHIST\COURSE1\LESSON1\REV
1.1.1.b Review
C:\USHIST\COURSE1\LESSON1\TEST
1.1.1.c Test
(b)

Figure 3. Sample outline (a) and corresponding directory tree (b).

CATS COURSEWARE FILE DESCRIPTIONS

The previous section described how to outline and organize course material using an MS- DOS directory tree on the computer's hard disk. In this section the underlying files used to create and present this course material are discussed.

There are two types of courseware files:

1. **Source files** are computer files used to create and maintain CBI using CATS. Storyboard files, graphic files, animation files, CATS variable files, and batch files are source files.

2. **Presentation files** are computer files used to present CBI using CATS. Segment files, profile files, DIR files, graphic files, and animation files are presentation files.

Note: Graphic files and animation files are both source and presentation files.

Storyboard Files

Purpose of Storyboard Files

Storyboard files are source files that contain the basic building blocks of the lesson. They contain information that controls:

1. **What the student sees on the screen during the CBI.** One storyboard file may contain several screens of information. For example, a storyboard file that presents a question might contain a question screen, a feedback screen for correct answers, and a feedback screen for incorrect answers.

2. **How the CBI will react to student responses.** For example, a storyboard may branch to a feedback screen if the student gives an inccrrect answer to a question.

3. **The lesson support files used by the CBI.** For example, a storyboard may display the contents of a graphic file on the screen.

4. **What data are collected by the CBI.** For example, a storyboard may save test scores in a text file for later use by the instructor.

How Storyboard Files Are Created

Storyboard files are text files that are created using a word processor (also called a text editor). The CATS user's manual contains detailed information about how to create storyboard files.

Where Storyboard Files Are Located

Storyboard files are located in subdirectories at the bottom of the courseware's directory tree.

Figure 4 illustrates a portion of the directory tree from Figure 1, showing storyboard files for question 1 through question 10.

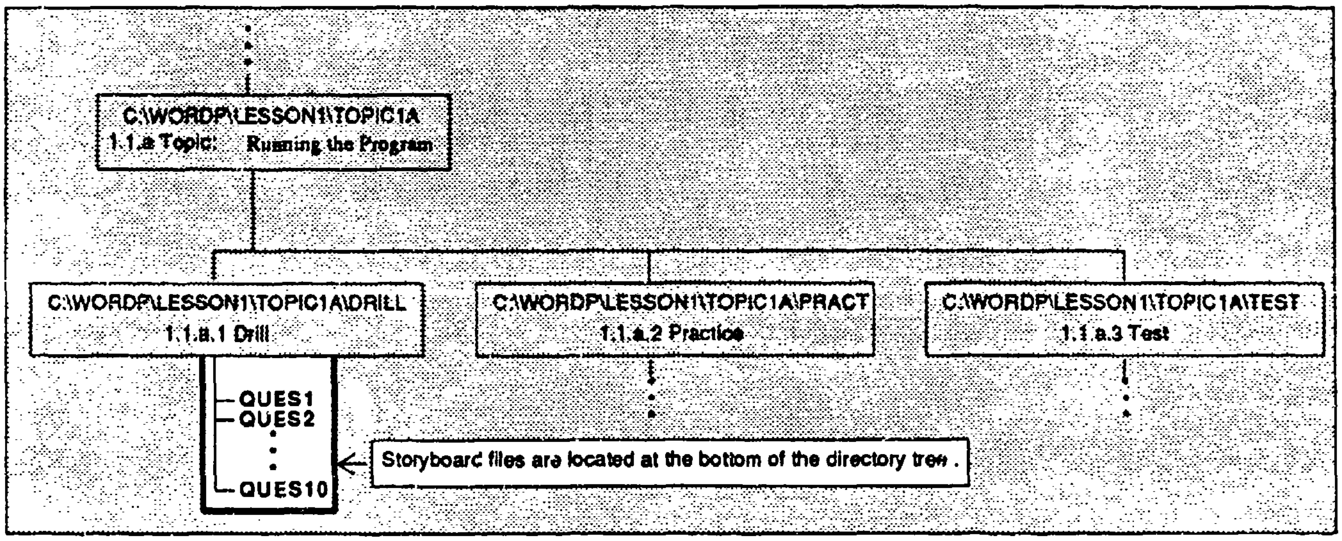

C:\WORDP\LESSON1\TOPIC1A
1.1.a Topic: Running the Program
C:\WORDP\LESSON1\TOPIC1A\DRILL
1.1.a.1 Drill
C:\WORDP\LESSON1\TOPIC1A\PRACT
1.1.a.2 Practice
C:\WORDP\LESSON1\TOPIC1A\TEST
1.1.a.3 Test
QUES1
QUES2
QUES10
Storyboard files are located at the bottom of the directory tree.

Figure 4. Storyboard file location for question 1 through question 10.

Graphic Files

Purpose of Graphic Files

Graphic files are both source files and presentation files. Graphic images are used by storyboard files to enhance CBI. Graphic files contain graphic images that can be displayed on a computer screen.

Two types of graphic files are used by the CATS: Full-screen files and cut files. Full-screen files contain graphic images that are displayed on a full computer screen. These files automatically have a suffix of `.PIC`. Cut files contain graphic images that are displayed on a part of the computer screen. These files automatically have a suffix of `.CUT`.

The CATS user's manual contains more detailed information about how to display graphic images in CATS courseware (see the Displaying Screen Images section).

How Graphic Files Are Created

Graphic files are produced using a graphics software program (also called a graphics editor) such as "Dr. Halo" or "PC-Paintbrush."

Where Graphic Files Are Located

Graphic files reside in the same directories as the storyboard files that use them. For example, if storyboard `QUES1` displays a full-screen graphic called `QUES1.PIC`, then that graphic file should reside in the same directory as `QUES1`.

Figure 5 shows the locations of the graphic files used by the storyboards for question 1 through question 10 in the word processing lesson. (In this figure and in some later figures, the list of graphic files is separated from the list of storyboard files. This separation is used to make the figure easier to understand. There is really no separation of file lists in a directory on the hard disk.)

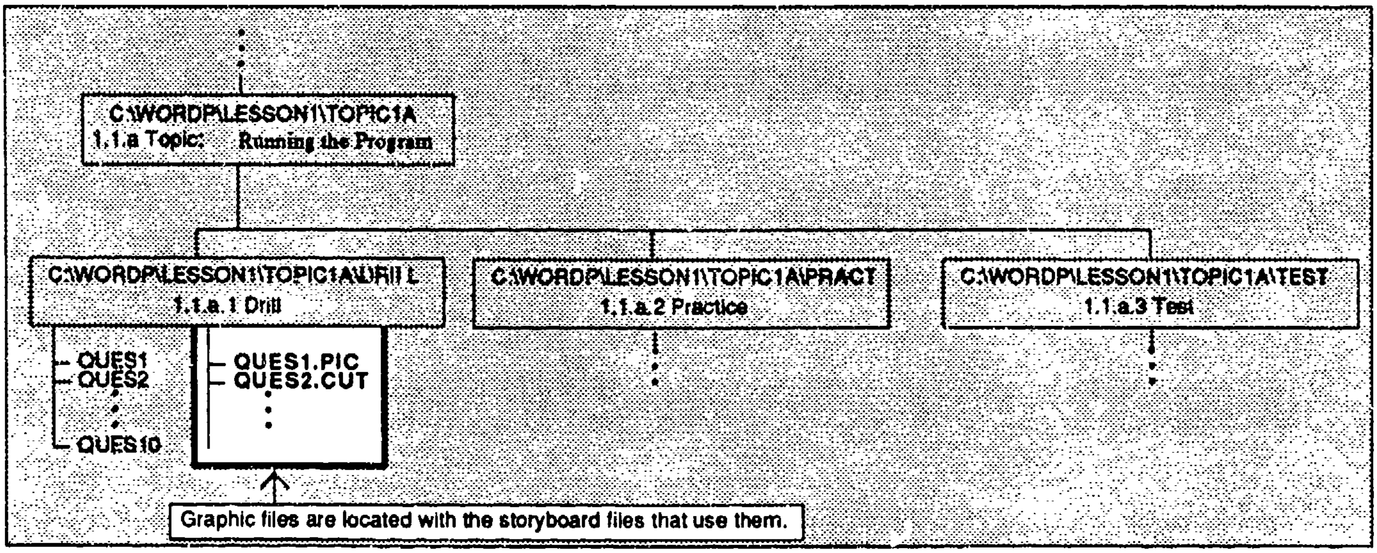

C:\WORDP\LESSON1\TOPIC1A
1.1.a Topic: Running the Program
C:\WORDP\LESSON1\TOPIC1A\DRILL
1.1.a.1 Drill
C:\WORDP\LESSON1\TOPIC1A\PRACT
1.1.a.2 Practice
C:\WORDP\LESSON1\TOPIC1A\TEST
1.1.a.3 Test
QUES1
QUES2
QUES10
QUES1.PIC
QUES2.CUT
Graphic files are located with the storyboard files that use them.

Figure 5. Graphic file location for question 1 through question 10.

Animation Files

Purpose of Animation Files

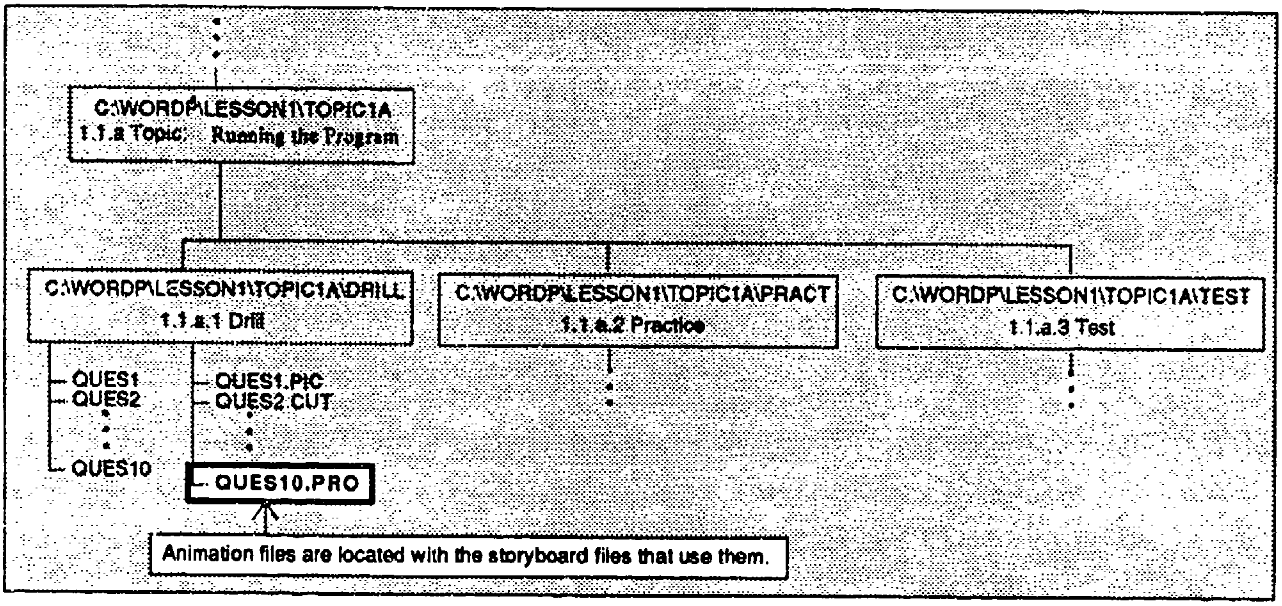
Animation files are both source and presentation files and contain animated graphic sequences. The file name extensions for these files are either `.PRO` or `.TXT`.

How Animation Files Are Created

Animation files are created using an animation software program (also called an animation editor) such as "Show Partner" or "Grasp."

Where Animation Files Are Located

Like graphic files, animation files reside in the same directories as the storyboard files that use them. Figure 6 shows the locations of the animation and graphic files used by the storyboards for question 1 through question 10 under one topic in the word processing lesson.


C:\WORDP\LESSON1\TOPIC1A
1.1.a Topic: Running the Program
C:\WORDP\LESSON1\TOPIC1A\DRILL
1.1.a.1 Drill
C:\WORDP\LESSON1\TOPIC1A\PRACT
1.1.a.2 Practice
C:\WORDP\LESSON1\TOPIC1A\TEST
1.1.a.3 Test
QUES1
QUES2
QUES10
QUES1.PIC
QUES2.CUT
QUES10.PRO

Animation files are located with the storyboard files that use them.

Figure 6. Animation file location for question 10.

CATS Variable File (CVARS)

Purpose of the CATS Variable File

By convention, the CATS variable file is named CVARS. CVARS is a source file and is an ASCII text file that defines the variables that are used by the lesson.

Every author variable used by the courseware in the courseware directory tree must be defined in the CVARS file. Whenever a variable is added or removed from any storyboard in the courseware directory tree, CVARS must be updated to reflect the change, and all the courseware in the directory tree must be recompiled. There must be only one CATS variable file (CVARS) for the courseware directory tree. For more information about the CATS variable file, see the CATS user's manual (Variables File section).

How the CATS Variable File is Created

CVARS is a text file created with a word processor or a text editor.

Where the CATS Variable File is Located

CVARS is always located in the topmost directory of the courseware directory tree. In Figure 7, the CVARS file is located in the WORDP directory at the top of the courseware directory tree.

Batch Files

Purpose of Batch Files

Batch files ease the process of building segment files (i.e., comprised versions of storyboard files) and presentation disks. They automatically execute the DOS commands that build a segment file or a set of presentation disks. Batch files are source files.

Batch file names have a suffix of .BAT.

The three types of batch files used in the lesson maintenance system are:

BUILD.BAT Compiles every storyboard in the directory and all its subdirectories and produces a package of all segment files in the directory and all its subdirectories.

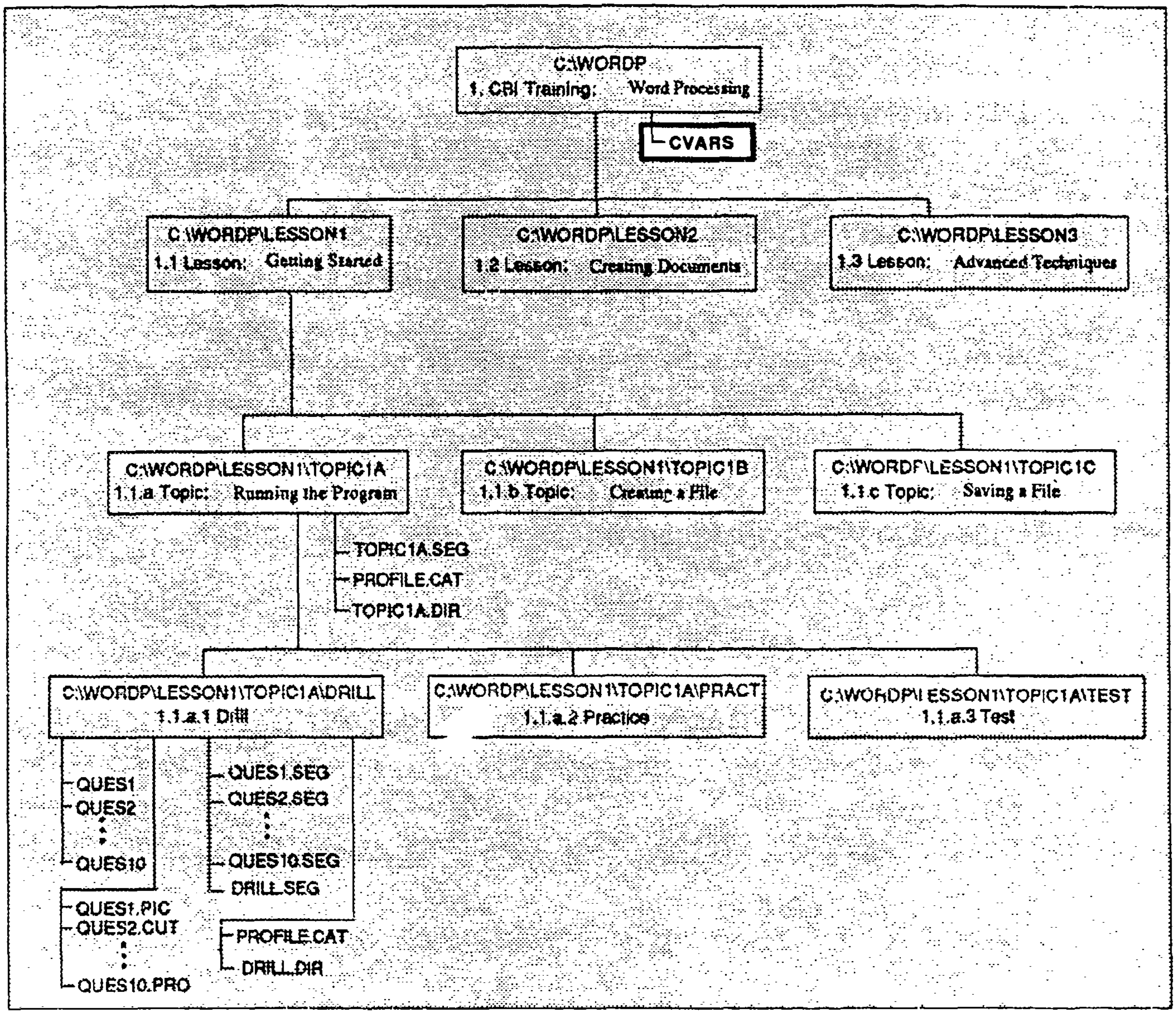


Figure 7. CATS variable file (CVARS) location in a sample courseware directory tree.

`QBUILD.BAT`	Uses already-compiled segment files to produce a package of all the segment files in a directory and all its subdirectories. This batch file will not compile any storyboards in the directory or its subdirectories.
`DISTRIB.BAT`	Creates a set of presentation disks for the courseware in a directory.

Appendix A contains detailed information about how the `BUILD.BAT` and `QBUILD.BAT` batch files work. Appendix B contains detailed information about how the `DISTRIB.BAT` batch file works.

How Batch Files Are Created

Batch files are text files created with a word processor or text editor.

Where Batch Files Are Located

BUILD.BAT and QBUILD.BAT reside in every directory of the courseware directory tree.

Figure 8 shows a portion of the courseware directory tree for the word processing example. BUILD.BAT and QBUILD.BAT batch files are located in every directory.

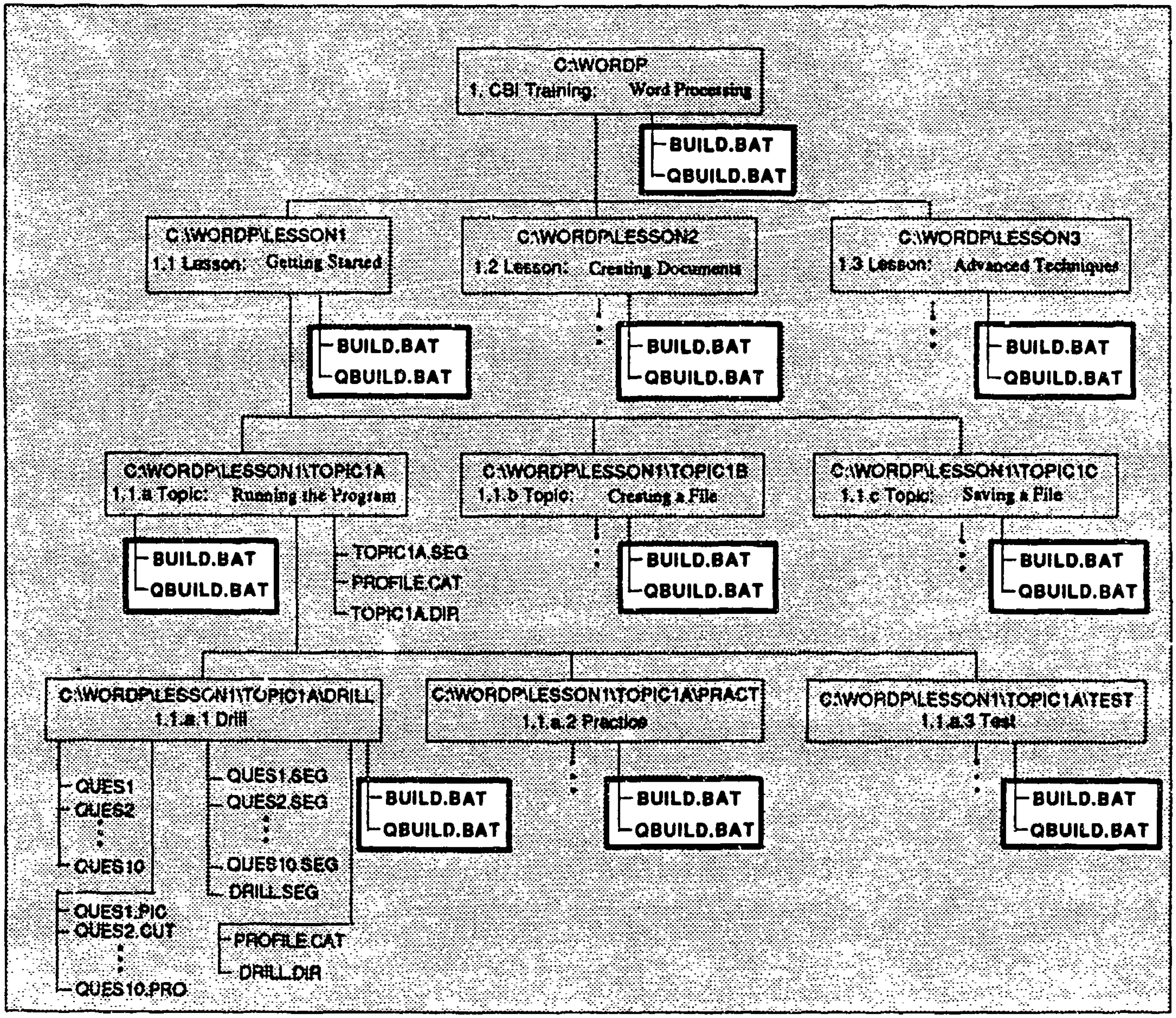


Figure 8. BUILD.BAT and QBUILD.BAT file locations in a sample directory tree.

The author makes courseware distributable by placing a `DISTRIB.BAT` file in each directory that contains the courseware to be distributed.

In Figuꞅe 9, a `DISTRIB.BAT` file is located in the top directory and in the directories containing courseware for lesson 1 through lesson 3. As a result, the word processing lessons could be distributed **together** by building presentation disks in the top directory, or each lesson could be distributed **separately** by building presentation disks in each lesson directory.

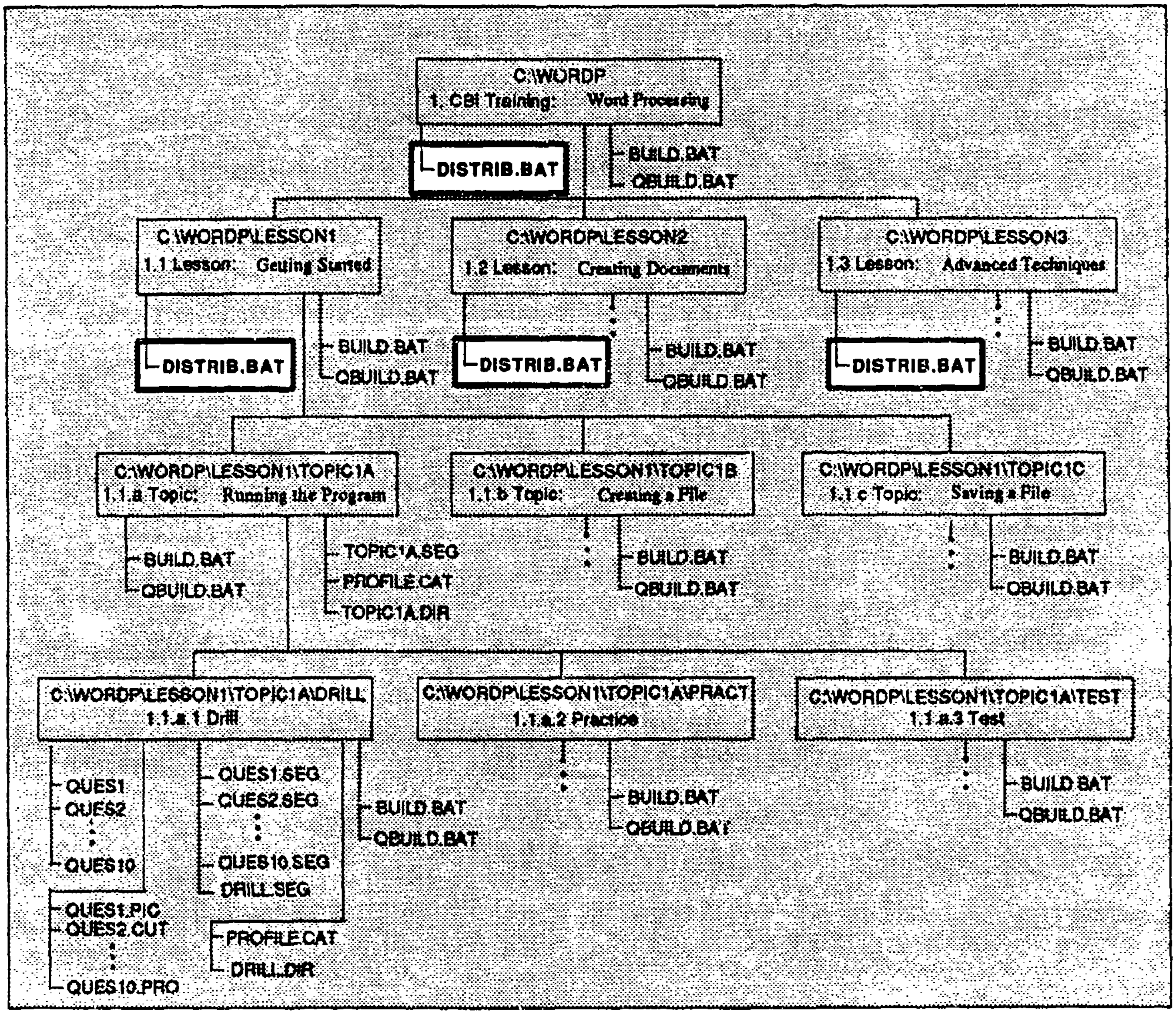


Figure 9. `DISTRIB.BAT` file locations in a sample directory tree.

Segment Files

Purpose of Segment Files

Segment files are presentation files that compress storyboard files into a single larger file. Two CATS software programs are used to create presentation files: CATSC, which compiles storyboard files into segment files that are in computer language, and CATSVIEW, which interprets the information stored in segment files for viewing. Segment files are automatically named with a .SEG suffix.

How Segment Files Are Created

Segment files are created by the CATS compiler (CATSC). Courseware developers or maintenance operators run each storyboard file through CATSC to check for errors and to produce a segment file for presentation.

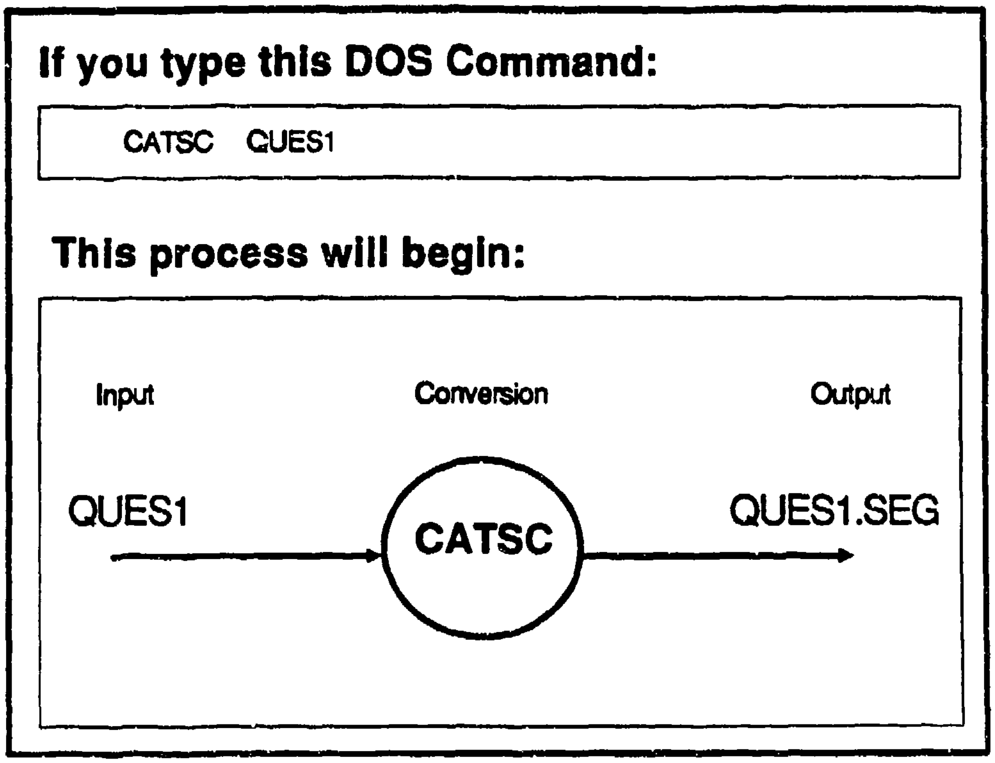
Figure 10 illustrates how a single segment file is produced from a single storyboard file. The shaded box represents the CATS compiler (CATSC). The arrow pointing to the box represents the storyboard file that is input to the compiler. The arrow pointing away from the box represents the segment file that is created by the compiler. At the top of the figure is the command that would be used to create the segment file. In this example, a storyboard called QUES1 is used to produce a question segment called QUES1.SEG. In this example, a segment name is not specified in the command. As a result, the storyboard file named QUES1 is used to create the segment file named QUES1.SEG.

Figure 11 shows how multiple segment files can be compiled to produce a single larger segment file. The larger segment contains all of the compressed information from the smaller segments. This is a convenient way of packaging many small lesson segments into a single larger segment file. In this example, all the information in segments QUES1.SEG through QUES10.SEG is packaged into one segment file. In this case, the name DRILL was specified in the command. Therefore, the CATS compiler names the resulting package segment DRILL.SEG.

The CATS user's manual includes more details about how to use CATSC to create segment files (see Compiling and Viewing a Lesson section).

Where Segment Files Are Located

There should be one segment file for each storyboard file in a directory. There should be an additional segment file that is a package containing all storyboard segments in each directory and all segments from the directory's subdirectories.

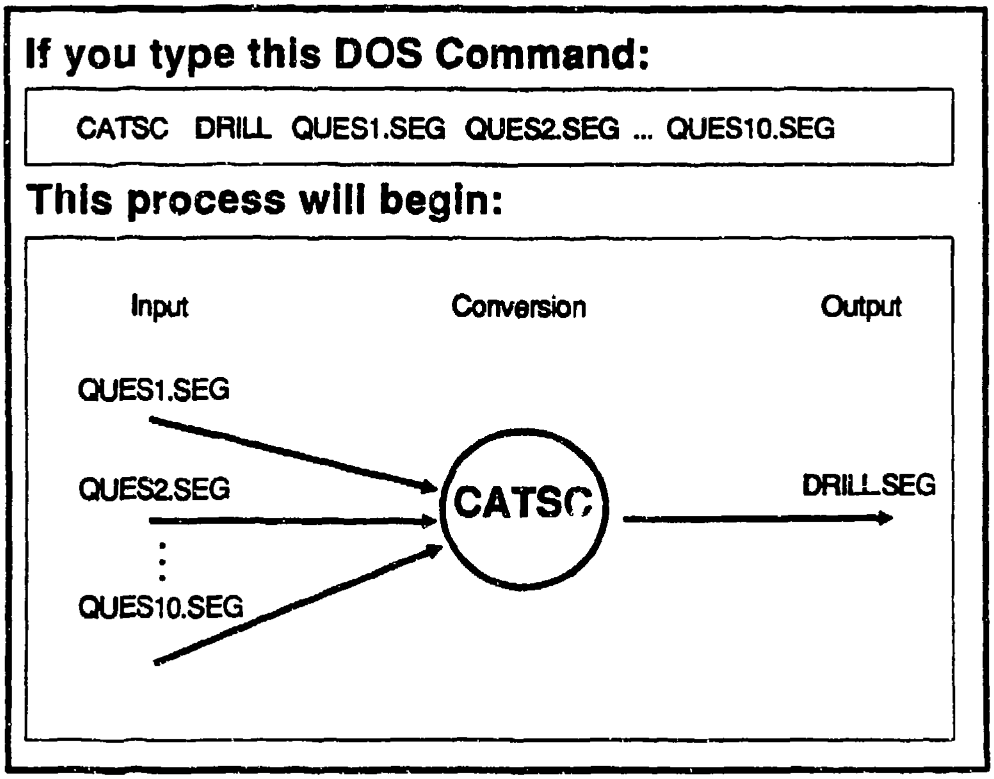


If you type this DOS Command:

```
CATSC   QUES1
```

This process will begin:

Figure 10. Producing segment files with the CATS compiler (CATSC).



If you type this DOS Command:

```
CATSC  DRILL  QUES1.SEG  QUES2.SEG ... QUES10.SEG
```

This process will begin:

Figure 11. Producing package segment files with CATSC.

The DRILL directory in Figure 12 contains one segment file for each storyboard file. It also contains the additional segment file DRILL.SEG that is a package of all the storyboard segments in the directory.

The TOPIC1A directory contains the segment TOPIC1A.SEG. This is a package that contains DRILL.SEG from the DRILL subdirectory, PRACT.SEG from the PRACT subdirectory, and TEST.SEG from the TEST subdirectory.

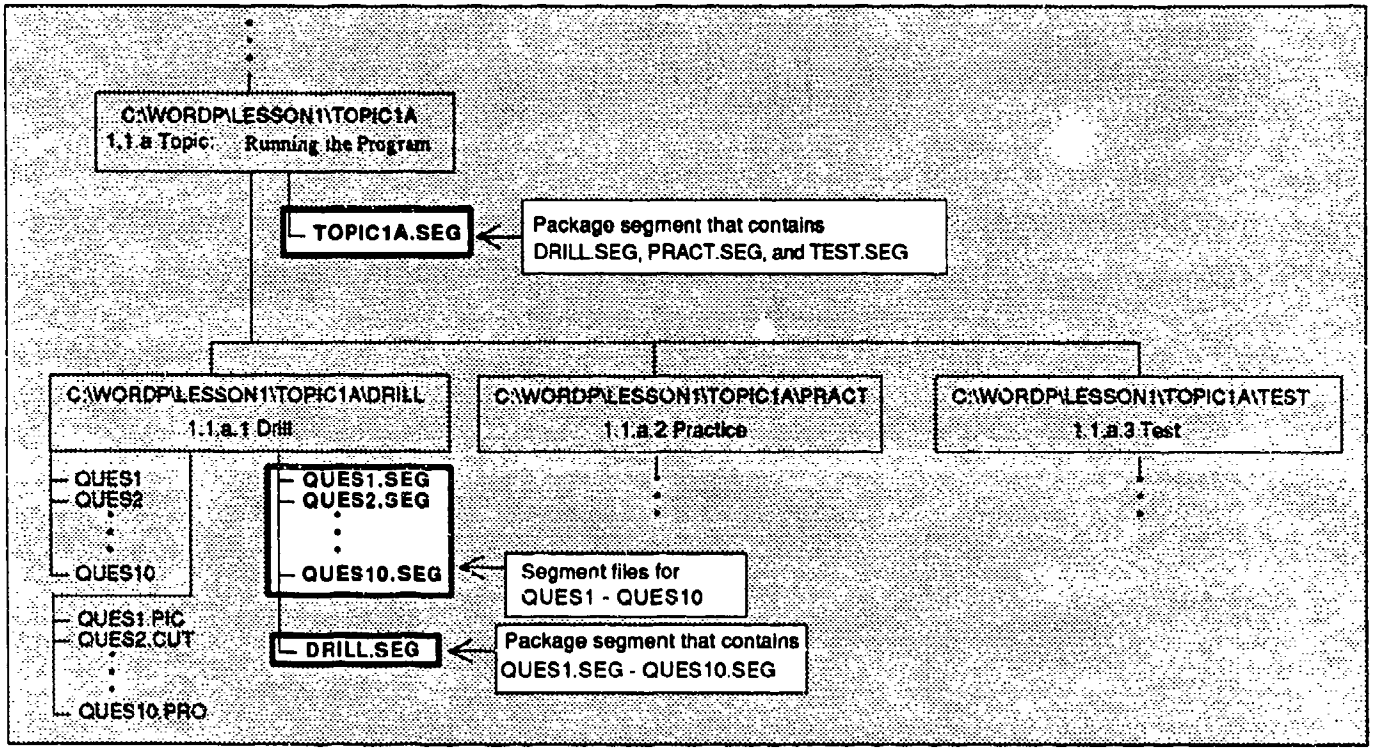

C:\WORDP\LESSON1\TOPIC1A
1.1.a Topic: Running the Program
TOPIC1A.SEG
Package segment that contains DRILL.SEG, PRACT.SEG, and TEST.SEG
C:\WORDP\LESSON1\TOPIC1A\DRILL
1.1.a.1 Drill
C:\WORDP\LESSON1\TOPIC1A\PRACT
1.1.a.2 Practice
C:\WORDP\LESSON1\TOPIC1A\TEST
1.1.a.3 Test
QUES1
QUES2
QUES10
QUES1.SEG
QUES2.SEG
QUES10.SEG
Segment files for QUES1 - QUES10
QUES1.PIC
QUES2.CUT
QUES10.PRO
DRILL.SEG
Package segment that contains QUES1.SEG - QUES10.SEG

Figure 12. How segment files can be used to package many small lesson segments into a single larger segment.

Profile Files

Purpose of Profile Files

Profile files are presentation files. The purpose of a profile file is to tell the CATSVIEW program which segment file to use when presenting CBI. To find the correct segment file, CATSVIEW will always look for a file called PROFILE.CAT and then read its contents. The contents of PROFILE.CAT identify which segment file CATSVIEW uses to present CBI. Profile files are always named PROFILE.CAT by CATSC.

How Profile Files Are Created

CATSC (the CATS compiler) creates a PROFILE.CAT file whenever it builds a segment file. The profile file contains the name of the most recent segment file that CATSC built in the directory. If CATSC is used twice in the same directory to produce two separate segment files, the profile file that identifies the segment that was built first will be overwritten by the profile file that identifies the segment file that was built second.

Where Profile Files Are Located

A PROFILE.CAT file resides in the same directory as the segment files it identifies.

If a profile file is in a directory that contains more than one segment, the profile file will identify the most recently created segment. There is only one PROFILE.CAT file in each directory.

Figure 13 shows the locations of the profile files that identify the most recently built segment files in the TOPIC1A directory and the DRILL directory.

DIR Files

Purpose of DIR Files

DIR files are presentation files that are used by CATSVIEW to locate information in a segment file. DIR file names always end with a .DIR extension.

How DIR Files Are Created

DIR files are automatically created the first time CATSVIEW is executed. If the DIR file is present, CATSVIEW will start faster. If the DIR file has been removed or is out of date, it will be built again the next time CATSVIEW is executed.

Where DIR Files Are Located

DIR files are created in the directories where a segment file is presented with CATSVIEW. In Figure 14, the TOPIC1A directory contains a DIR file called TOPIC1A.DIR. This DIR file was created when CATSVIEW presented TOPIC1A.SEG. There is also a DIR file called DRILL.DIR in the DRILL directory. This DIR file was created when DRILL.SEG was presented with CATSVIEW.

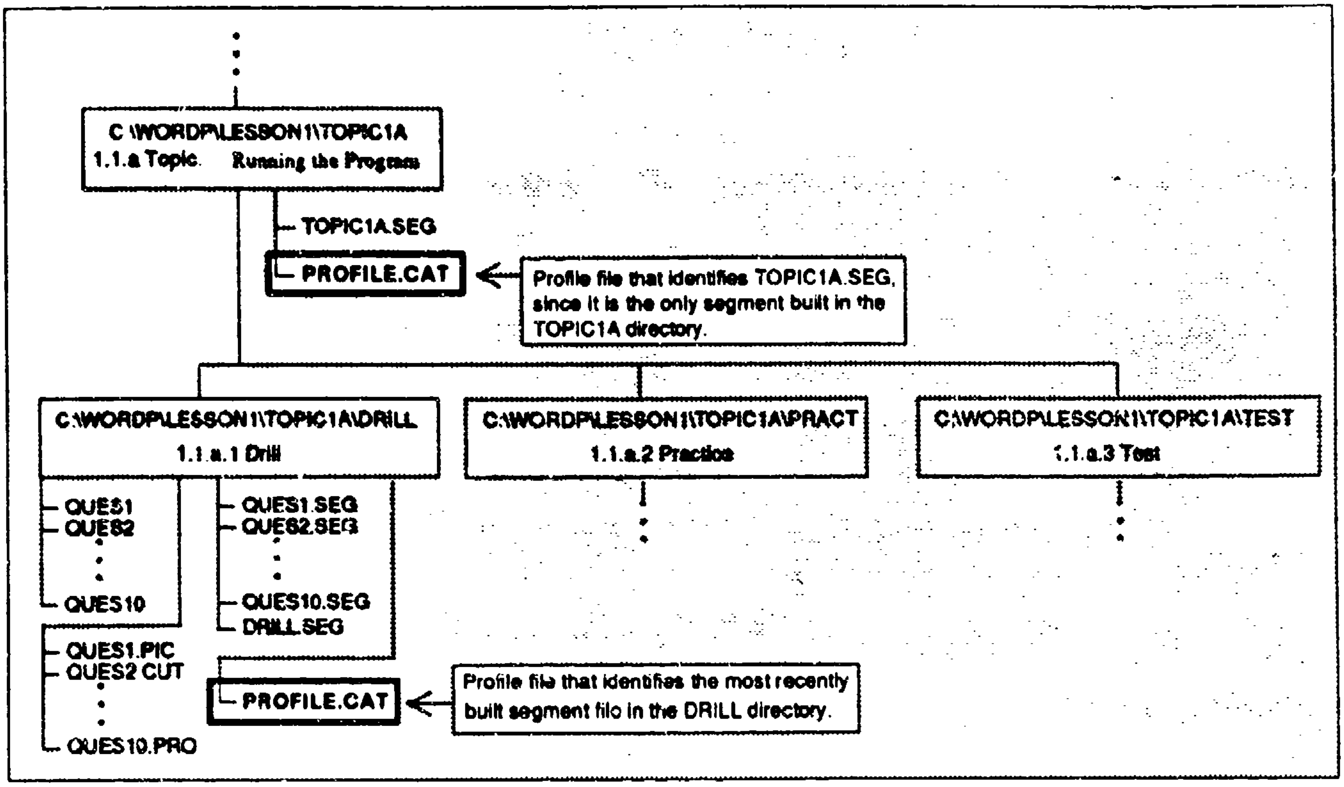

C:\WORDP\LESSON1\TOPIC1A
1.1.a Topic: Running the Program
TOPIC1A.SEG
PROFILE.CAT
Profile file that identifies TOPIC1A.SEG, since it is the only segment built in the TOPIC1A directory.
C:\WORDP\LESSON1\TOPIC1A\DRILL
1.1.a.1 Drill
C:\WORDP\LESSON1\TOPIC1A\PRACT
1.1.a.2 Practice
C:\WORDP\LESSON1\TOPIC1A\TEST
1.1.a.3 Test
QUES1
QUES2
QUES10
QUES1.SEG
QUES2.SEG
QUES10.SEG
DRILL.SEG
QUES1.PIC
QUES2.CUT
QUES10.PRO
PROFILE.CAT
Profile file that identifies the most recently built segment file in the DRILL directory.

Figure 13. How profile files reside in the same directory as the segment files they identify.

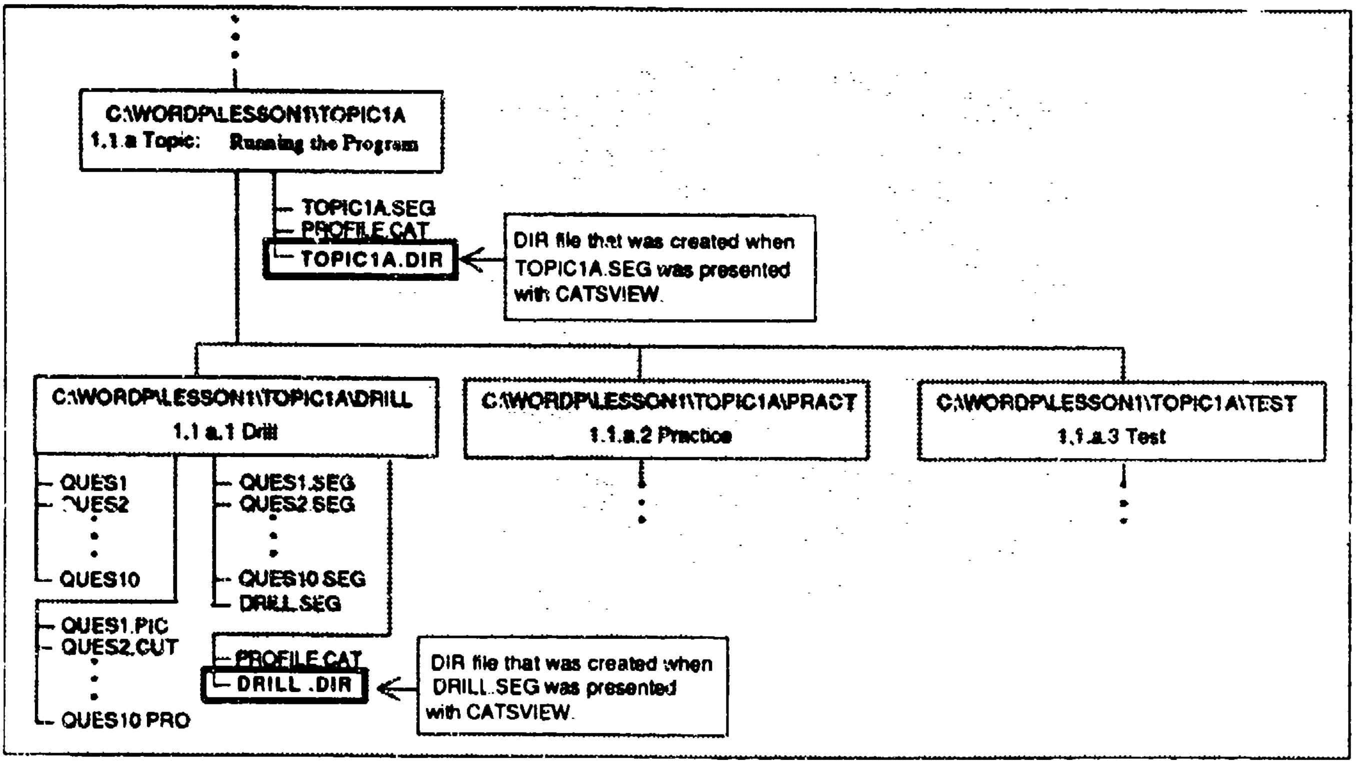

C:\WORDP\LESSON1\TOPIC1A
1.1.a Topic: Running the Program
TOPIC1A.SEG
PROFILE.CAT
TOPIC1A.DIR
DIR file that was created when TOPIC1A.SEG was presented with CATSVIEW.
C:\WORDP\LESSON1\TOPIC1A\DRILL
1.1.a.1 Drill
C:\WORDP\LESSON1\TOPIC1A\PRACT
1.1.a.2 Practice
C:\WORDP\LESSON1\TOPIC1A\TEST
1.1.a.3 Test
QUES1
QUES2
QUES10
QUES1.SEG
QUES2.SEG
QUES10.SEG
DRILL.SEG
QUES1.PIC
QUES2.CUT
QUES10.PRO
PROFILE.CAT
DRILL.DIR
DIR file that was created when DRILL.SEG was presented with CATSVIEW.

Figure 14. DIR files are created in the directories where a segment file is presented.

LESSON MAINTENANCE SYSTEM PROCEDURES

This section describes the lesson maintenance system procedures that allow CATS courseware instructional developers to install the courseware onto the lesson maintenance system, to create presentation disks, and to change CATS courseware.

As the courseware developer, you probably received the CATS courseware on floppy disks and are now responsible for:

- Installing the courseware onto the lesson maintenance system.
- Distributing the courseware to students.
- Changing the courseware to meet new requirements requested by instructors or students.

This document is intended to help you accomplish this with as little trouble as possible. First, we'll discuss the materials with which you'll be working, then we'll discuss the steps of installing, distributing, and changing the courseware.

THE MATERIAL YOU HAVE

You most likely received the courseware in several sets of floppy disks:

1. **Presentation** disks used to present courseware to students. (These disks are probably labeled "Disk 1," "Disk 2," etc.)

2. **Source** disks that contain the source files used to develop the courseware. (These disks are probably labeled "Source Files" or "Storyboard Files" and name the courseware involved.)

You probably also received a hardcopy courseware directory for each lesson illustrating the courseware directory tree that is used to organize the courseware files. The courseware directory will help you visualize the organization of the courseware files on the source and presentation disks and in your lesson maintenance system.

After receiving the source disks, you will most likely want to install the source files onto your lesson maintenance system's hard disk. The next section, Installing the Courseware Files, covers the procedures for doing this.

Next, you may want to distribute courseware. This means that you will have to make copies of your presentation disks. The procedures for copying and updating these disks are discussed in the section ahead called Making Presentation Disks.

Eventually you may need to make a variety of changes to the courseware. Courseware content may change, an instructor may want to shift emphasis from one point to another, or you may want to add or delete information. Changing courseware is probably the most complicated task you will have to do. The procedures for changing courseware are in the Changing Courseware section.

INSTALLING THE COURSEWARE FILES

The courseware directory illustrates the courseware directory tree used to organize the source files in the lesson maintenance system.

For each directory in the courseware directory tree, there should be one or more source disks that contain all the files belonging in that directory.

To install the source files from the source disks to your hard disk, do the following:

1. Build the courseware directory tree that is used to organize the source files. Do this by using the DOS `mkdir` command. You can use the courseware directory as a "blueprint" of this directory tree.

2. After the courseware directory tree is built, place all the source files in their proper directories. Each source file disk should be labeled with a directory name. All the files on a source file disk belong in the directory that is on the disk's label. Figure 15 illustrates how the label on the source file disk indicates the directory where all the files on the disk belong.

Therefore, copy all files of each source file disk into the directory indicated by the disk's label.

MAKING PRESENTATION DISKS

Presentation disks contain all the courseware files needed to **present** the CBI. The following sections discuss:

1. How to make copies of presentation disks you already have.

2. How to update presentation disks to reflect changes made to the courseware.

3. What to do to distribute a portion of the courseware that was not initially made distributable.

Making Copies of Presentation Disks

If the courseware has not changed, and if you want to copy some presentation disks, use the DOS `DISKCOPY` command to make the copies. (Check your DOS manual for specific instructicns on how to use the `DISKCOPY` command.)

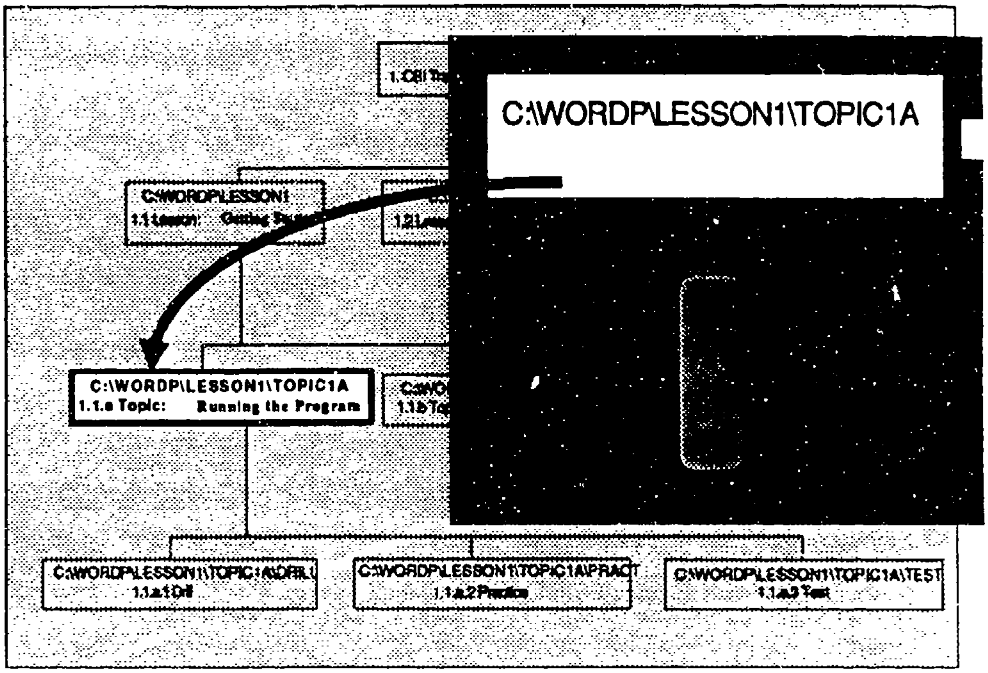

C:\WORDP\LESSON1\TOPIC1A
C:\WORDP\LESSON1
1.1 Lesson: Getting Started
C:\WORDP\LESSON1\TOPIC1A
1.1.a Topic: Running the Program
C:\WORDP\LESSON1\TOPIC1A\DRILL
1.1.a.1 Drill
C:\WORDP\LESSON1\TOPIC1A\PRACT
1.1.a.2 Practice
C:\WORDP\LESSON1\TOPIC1A\TEST
1.1.a.3 Test

Figure 15. Example of how the label on a source disk can indicate the directory where all files on the disk belong.

Updating Presentation Disks

To distribute courseware that has been **changed**, you must update its set of presentation disks. (For procedures on how to change courseware, see the Changing Courseware section.)

Each set of presentation disks you received corresponds to a particular directory in the courseware directory tree. Each of these directories should contain the `DISTRIB.BAT` batch file that is used to build presentation disks.

For example, Figure 16 illustrates a portion of the directory tree for the word processing example. You would have received a set of presentation disks for the courseware in the directories `\WORDP` and `LESSON1` through `LESSON3`, since these are the directories that contain the batch files `DISTRIB.BAT`.

Updating presentation disks consists of using `DISTRIB.BAT` to rebuild the presentation disks. The procedure for updating presentation disks is given in Table 1.

Note: Before using `DISTRIB.BAT` to update the presentation disks, make sure all of the segment files are current.

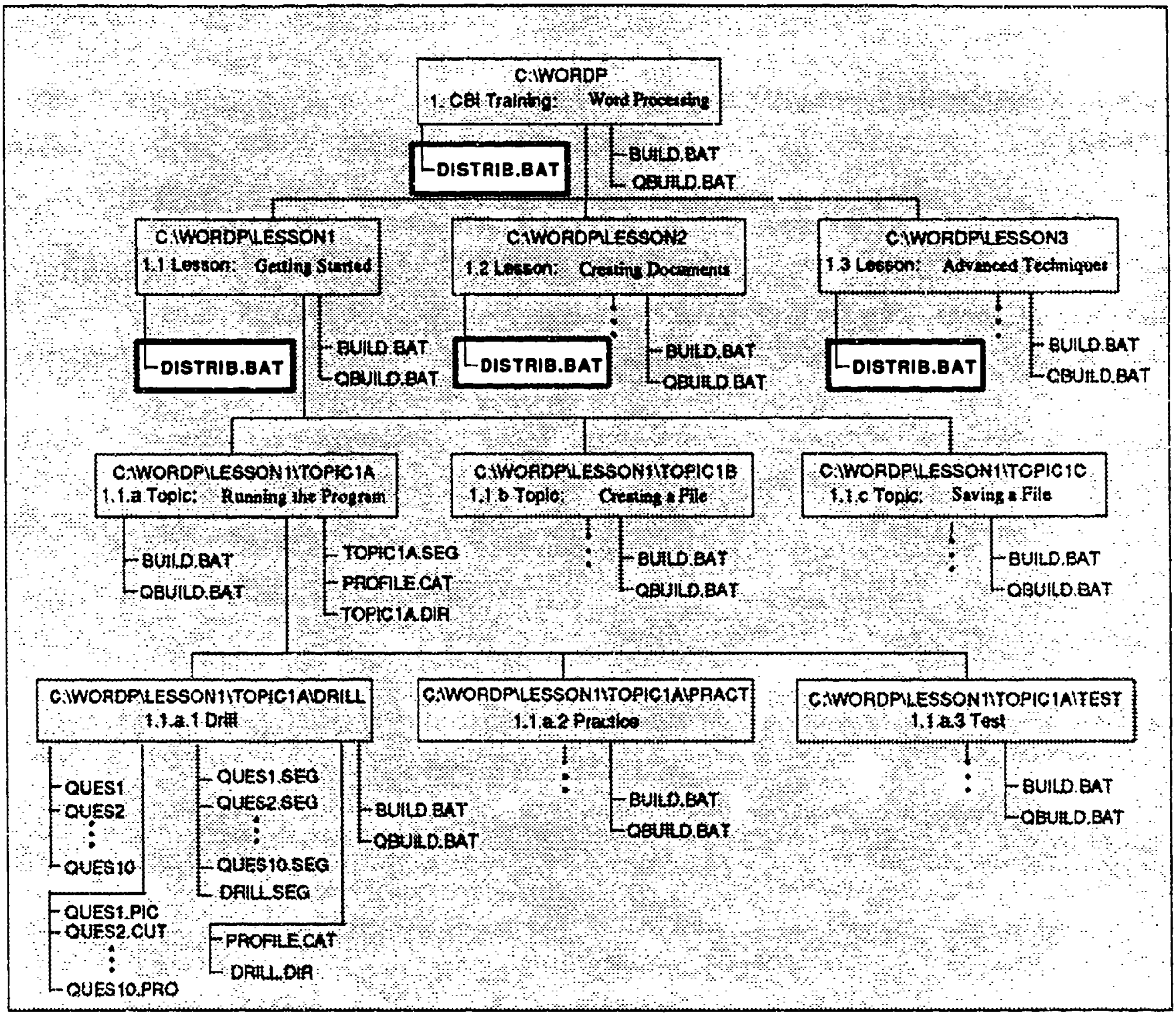


Figure 16. Courseware in directories that contain the `DISTRIB.BAT` file can be distributed.

`DISTRIB.BAT` does not build any segment files. It builds only presentation disks. This means that if you did not rebuild a segment file after changing a storyboard file and used `DISTRIB.BAT` to build presentation disks, then the courseware on the presentation disks would not reflect any of the changes you made to the storyboard files. (See Appendix B for details of how `DISTRIB.BAT` works.)

Table 1

Updating Presentation Disks

Procedure For Updating Presentation Disks	
1. Movc to the appropriate directory.	• Move to the directory in the courseware directory tree whose presentation disks need to be updated.
2. Type `DISTRIB.BAT`	• Follow the instructions on the screen. Notice that `DISTRIB.BAT` should: • Format the presentation disks (including a generic student disk). • Copy the segment files, the graphic and animation files, and the CATS presentation software files to the presentation disks.

Making Courseware Distributable

Any courseware in the courseware directory tree can become distributable. The presentation disks you initially received correspond to the portions of the courseware the original developer decided to make distributable. This section discusses how you can make other portions of the courseware distributable.

To make a portion of the courseware distributable, follow the steps in Table 2.

CHANGING COURSEWARE

You will probably want to make changes to the courseware from time to time. Lesson content may change or an instructor may want to shift emphasis from one point to another. (This document covers only those actions that occur before and after a storyboard file is changed. It does not focus on the process of changing a storyboard file. The CATS user's manual offers detailed information about how to create and update a storyboard file.)

A Helpful Hint: Back Up Your Courseware Directory Tree

Before beginning to change the courseware, it is a good idea to make a backup copy of the existing courseware directory tree and all the source files within it. There are two reasons for this action. First, if you make a mistake when changing courseware, you will be able to use your backup copy to start from scratch. Second, if your hard disk is erased (or something else equally damaging happens), you will be able to rebuild the entire courseware directory tree from the backup copy.

It is also a good idea to make a backup copy of the courseware directory tree when releasing any new versions of the courseware. Then, if you are asked to alter a previous release of the courseware, you can rebuild the appropriate version of the courseware directory tree.

You must determine the best way to make backup copies. The method you choose will depend on which software packages you already have and on the types of tools with which you are most comfortable. If you have no special backup tool, you can always use the DOS BACKUP and RESTORE facilities.

Table 2

Making a Portion of the Courseware Distributable

Procedures For Making A Portion Of The Courseware Distributable	
1. Create the DISTRI.BAT batch file for the part of the courseware you want distributed.	• Be sure you understand how the DISTRIB.BAT file works. (See Appendix B for detailed information detailed information about DISTRIB.BAT.) • Move to the directory that corresponds to the part of the courseware you want to make distributable. If any of the files in the directory are larger than 360 kilobytes, follow the instructions in Appendix C, Handling Presentation Disk Size Constraints. • Create a DISTRIB.BAT for the new directory. The easiest way to create a new DISTRIB.BAT file is to alter a copy of an existing DISTRIB.BAT file. Do this by finding an existing DISTRIB.BAT file and copying it into the directory. Use one that belongs to a similar directory.
2. Edit the new batch file.	• Edit the new DISTRIB.BAT file with your word processor and revise the DOS commands so they will: • Copy all the graphic and animation files from the directory and all its subdirectories into the temporary directory called INSTALL. (Be sure to use *.PIC, *.CUT, and *.PRO or *.TXT when copying the graphic and animation files to the INSTALL directory. That way, you will not have to update the DISTRIB.BAT each time you add or remove a graphic or animation file.) • Copy the directory's package segment into the INSTALL directory. • Save the DISTRIB.BAT file as an non-document or ASCII file.
3. Test the new batch file.	• Execute the new DISTRIB.BAT file to verify that it copied the directory's package segment as well as all the graphic and animation files for the directory and its subdirectories to the INSTALL directory. • Watch for compilation errors. • Use the presentation disks to install the courseware on a different computer and verify that the courseware works.

Making Changes to Courseware

When making changes, it is important to know which changes will require you to also update the BUILD and QBUILD batch files in the courseware directory tree. Table 3 should help.

Table 3

Types of Courseware Changes that DO or DO NOT Affect the BUILD and QBUILD Batch Files

	Must BUILD and QBUILD Batch Files Be Updated?	
Type Of Courseware Change	NO	YES
Alter a storyboard in a storyboard file	X	
Add storyboard to a storyboard file	X	
Remove a storyboard from a storyboard file	X	
Alter a graphic or animation file	X	
Add a new graphic or animation file	X	
Remove a graphic or animation file	X	
Add a new storyboard file		X
Remove a storyboard file		X
Add a new directory		X
Remove a directory		X

Notice that changes made to an existing file require no changes to the BUILD and QBUILD batch files. Also, adding or removing a graphic or animation file requires no changes to the batch files. However, if you add or remove a storyboard file, the BUILD and QBUILD batch files must be updated to reflect the change.

BUILD and QBUILD batch files contain DOS commands that refer to storyboard files. If a storyboard file has been added, the storyboard file's name must be added to the appropriate batch files. If a storyboard file has been removed, the storyboard file's name must be removed from the appropriate batch files.

If any of the files that you create or alter become larger than 360 kilobytes, follow the instructions given in Appendix C, Handling Presentation Disk Size Constraints.

If the changes involve adding or deleting author variables, you must update the CATS variable file (CVARS) and recompile all the courseware in the courseware directory tree.

Changes That DO NOT Affect BUILD and QBUILD Batch Files

The procedures for making changes that do not affect BUILD and QBUILD batch files are threefold. They consist of (1) making the change, (2) recompiling the segment, and then (3) testing the results. The procedure for altering a storyboard in a storyboard file is given in Table 4.

Table 4

Altering a Storyboard in a Storyboard File

Procedure For Altering a Storyboard in a Storyboard File	
1. Change the storyboard file.	• Move to the directory that contains the storyboard file. • Retrieve the storyboard file as a non-document or ASCII file. • Use your word processor to edit the storyboard file. • Save the changed storyboard file as a non-document or ASCII file.
2. Recompile the segment for the storyboard file.	• Use CATSC to rebuild the segment file for the storyboard file.. • Watch for any compilation errors. If there are such errors, reedit the storyboard file to fix them and then go back to the beginning of step 2 to rebuild the segment file. • Check the size of the segment file and any graphic or animation files that it requires. Graphic or animation files larger than 360 kilobytes are currently not distributable. If there are any files larger than 360 kilobytes, you may have to alter the DISTRIB.BAT file for the directory. Follow the instructions in Appendix C, Handling Presentation Disk Size Constraints.
3. Test the segment for the storyboard file.	• Use CATSVIEW to view the new segment file and verify that the storyboard reflects your alterations. • Verify that the segment file's screens, branching, data collection, etc., are correct.

When altering or adding a graphic or animation file, use a graphic or animation editor instead of a word processor. There is no need to rebuild segments if you have only changed an existing graphic or animation file. However, you should still use CATSVIEW to verify that the altered graphic or animation file is correct.

When you finish changing the storyboard files in a directory, use QBUILD.BAT or BUILD.BAT to rebuild the directory's package segment, and then use CATSVIEW to test it. You want to make sure that the changes made to each storyboard file will not conflict with other storyboard files in the directory.

Note: To use QBUILD.BAT, **all** storyboards must be already compiled with the most current segment files. If all storyboards are **not** compiled, use BUILD.BAT to create the updated segment files.

After you finish making changes between package segments (i.e., between subdirectories), rebuild the next highest package segment in the courseware directory tree. You want to be sure your changes have not caused conflicts between package segments.

Changes That DO Affect BUILD and QBUILD Batch Files

Changes that cause a storyboard file to be added, removed, or renamed will affect the BUILD and QBUILD batch files. These batch files contain commands that manipulate files in the directory tree. If a storyboard file has been added, the storyboard file name must be added to the appropriate batch files. If a storyboard file has been removed, the storyboard file name must be removed from the appropriate batch files. If a storyboard file has been renamed, the original name must be removed and the new name must be added to the appropriate batch files.

Adding or Removing a Storyboard File

The steps for adding a storyboard file are given in Table 5, and an example is illustrated in Figure 17. The steps for removing a storyboard file are given in Table 6.

Adding a Directory to the Courseware Directory Tree

Recall that each directory or storyboard file in the courseware directory tree corresponds to an item in the instructional outline of the courseware material. If you choose to add an item to the outline, you will have to add a directory and/or a storyboard file to the courseware directory tree.

If the new item belongs at the lowest level of the instructional outline (i.e., if it is not divided into other items), then you have only to add a storyboard file to the appropriate directory in the directory tree. Adding a question to the drill in the word processing example illustrates this type of change. If the new item is divided into other items, you must add a directory and the appropriate number of storyboard files to the courseware directory tree. Adding an item called "review" that is divided into two questions to lesson 1 in the word processing CBI is an example of this type of change and is illustrated in Figure 18.

The procedure for adding a directory to the courseware directory tree is presented in Table 7.

Removing a Directory from the Courseware Directory Tree

If you choose to remove an item from the instructional outline, you may have to remove a directory from the courseware directory tree. If the item you want to remove is not divided into other items, you have only to remove the storyboard file that corresponds to the item. Removing a question from the drill in the word processing example illustrates this type of change. However, if the item you want to remove is divided into other items, you must remove the directory and courseware files that correspond to the item. Removing the item called "drill" in the word processing example illustrates this type of change.

The procedure for removing a directory from the courseware directory tree is presented in Table 8.

Table 5

Adding a Storyboard File

Procedure for Adding a Storyboard File	
1. Create the new storyboard file.	• Move to the directory that will contain the new storyboard file. • Use your word processor to create the new storyboard file. Save the new storyboard file as a non-document or ASCII file.
2. Build the segment for the new storyboard file.	• Use CATSC to build a segment for the new storyboard file. • Watch for any compilation errors. If there are such errors, reedit the storyboard file and then go back to the beginning of step 2 to rebuild the segment. • Check the size of the segment file and any graphic or animation files that it requires. Graphic or animation files larger than 360 kilobytes are currently not distributable. If there are any files larger than 360 kilobytes, you may have to alter the DISTRIB.BAT file for the directory. Follow the instructions in Appendix C, Handling Presentation Disk Size Constraints.
3. Test the new segment.	• Use CATSVIEW to view the new segment and verify that its screens, branches, data collection, etc., are correct.
4. Locate every batch file affected by this change.	• Determine which batch files should be updated. (Step 4 focuses on locating the batch files that are affected by adding a storyboard file. Step 5 covers the changes that should be made to these batch files.) To locate the affected batch files: a. Begin in the directory where the new storyboard resides. The BUILD.BAT and QBUILD.BAT files in this directory must be updated. b. Move up one directory. The BUILD.BAT batch file in this directory must be updated. Continue to update the BUILD.BAT files until you reach the topmost directory in the CBI directory tree. Steps a and b define a path of directories beginning with the directory where the new storyboard resides and continuing to the topmost directory in the directory tree. Each BUILD.BAT file in this path must be updated. In addition, the QBUILD.BAT file in the directory where the storyboard resides must be updated. For example, Figure 17 illustrates the batch files at each level that must be updated if a storyboard called QUES11 is added to the directory DRILL. The highlighted area represents the path from the directory containing the new storyboard to the topmost directory in the CBI directory tree. Each BUILD.BAT file along this path must be updated. In addition, the QBUILD.BAT in the DRILL directory must be updated.

Table 5 (Continued)

Procedure for Adding a Storyboard File	
5. Update every batch file affected by this change.	• Update each batch file. To do this you must know how the batch file works. Appendix A describes in detail how BUILD.BAT and QBUILD.BAT work. For each BUILD.BAT that is affected you should: • Add a command that builds a segment file for the new storyboard file. • Add a command that places the storyboard file's segment into the package segment that resides in the same directory as the new storyboard file. For the QBUILD.BAT that is affected you should: • Add a command that places the storyboard file's segment into the package segment that resides in the same directory as the new storyboard file.
6. Test every batch file affected by this change.	• Execute each updated batch file and verify that its modifications, screens, data collection, etc., are correct. Check all branch paths in the updated segment files. Make sure that a storyboard does not branch to an incorrect or nonexistent event. • Executing each updated batch will also give you the opportunity to verify that the new storyboard file does not conflict with the rest of the courseware in the directory tree.

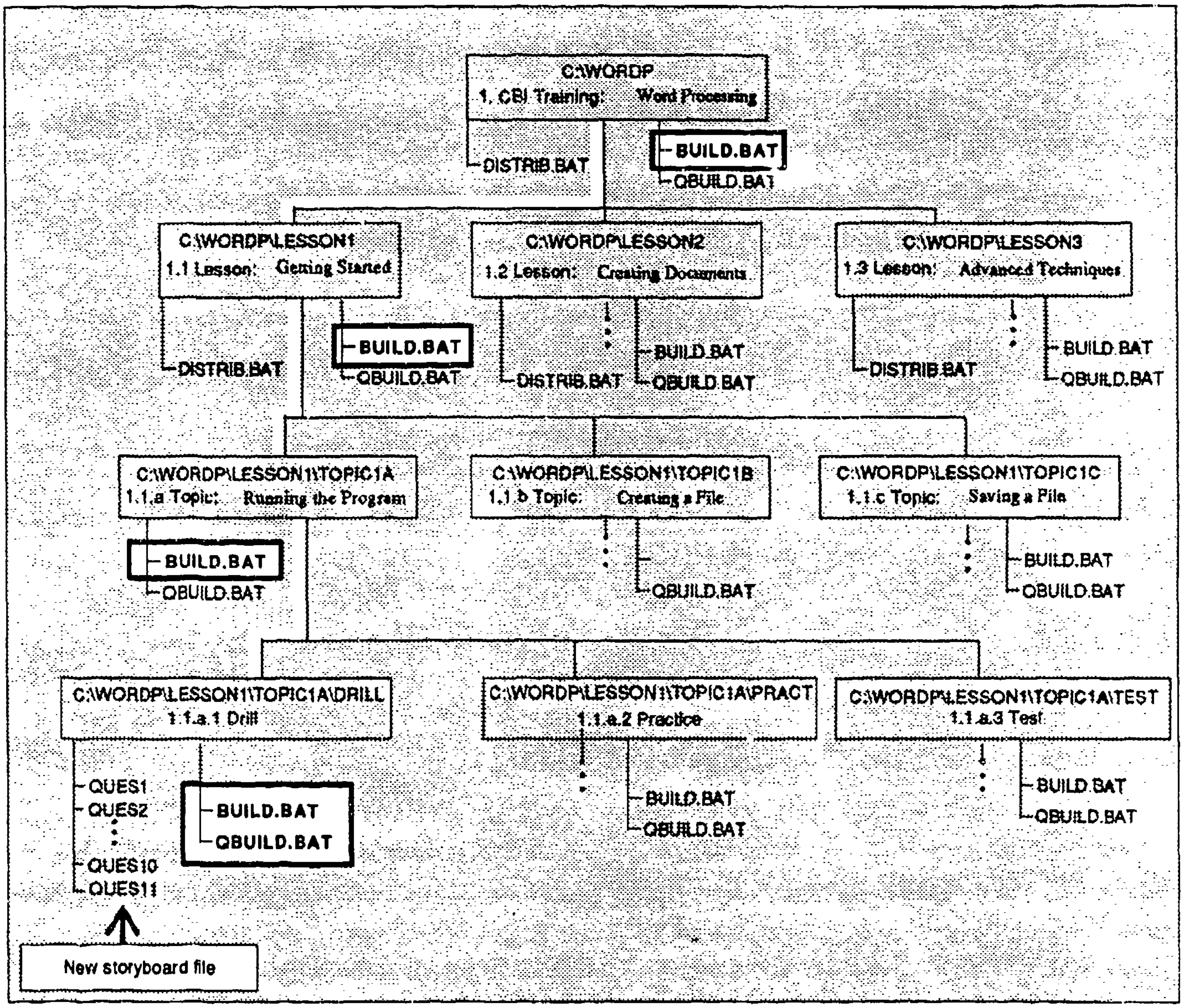


Figure 17. Each `BUILD.BAT` file, from the directory containing the new storyboard to the topmost directory in the CBI directory tree, must be updated.

Table 6

Removing a Storyboard File

	Procedure for Removing a Storyboard File
1. Remove the storyboard file.	• Move to the directory that contains the storyboard file. • Remove the storyboard file.
2. Locate every batch file affected by this change.	• Determine which batch files should be updated. (Step 2 focuses on locating the batch files that are affected by removing a storyboard file. Step 3 covers the changes that should be made to these batch files.) To locate the affected batch files: a. Begin in the directory where the storyboard file resided. The `BUILD.BAT` and `QBUILD.BAT` files in this directory must be updated. b. Move up one directory. The `BUILD.BAT` file in this directory must be updated. Continue to update the `BUILD.BAT` files until you reach the topmost directory in the CBI directory tree. Steps a and b define a path of directories beginning with the directory where the storyboard resides and continuing to the topmost directory in the CBI directory tree. Each `BUILD.BAT` file in this path must be updated. In addition, the `QBUILD.BAT` file in the directory where the storyboard resides must be updated.
3. Update every batch file affected by this change.	• Update each batch file. To update a batch file you need to know how the batch file works. See Appendix A for detailed information about the `BUILD.BAT` and `QBUILD.BAT` files. For each `BUILD.BAT` that is affected you should: • Remove the command that builds a segment for the storyboard file. • Remove the command that places the storyboard file's segment into the package segment that resides in the same directory as the storyboard file. For the `QBUILD.BAT` that is affected you should: • Remove the command that places the storyboard's segment into the package segment that resides in the same directory as the storyboard file.
4. Test every batch file affected by this change.	• Execute each updated batch file and verify that its modifications, screens, data collection, etc., are correct. Check all branch paths in the updated segment files. Make sure that a storyboard doesn't branch to an incorrect or nonexistent event. • Executing each updated batch will also give you the opportunity to verify that removing the storyboard file does not conflict with the rest of the courseware in the directory tree.

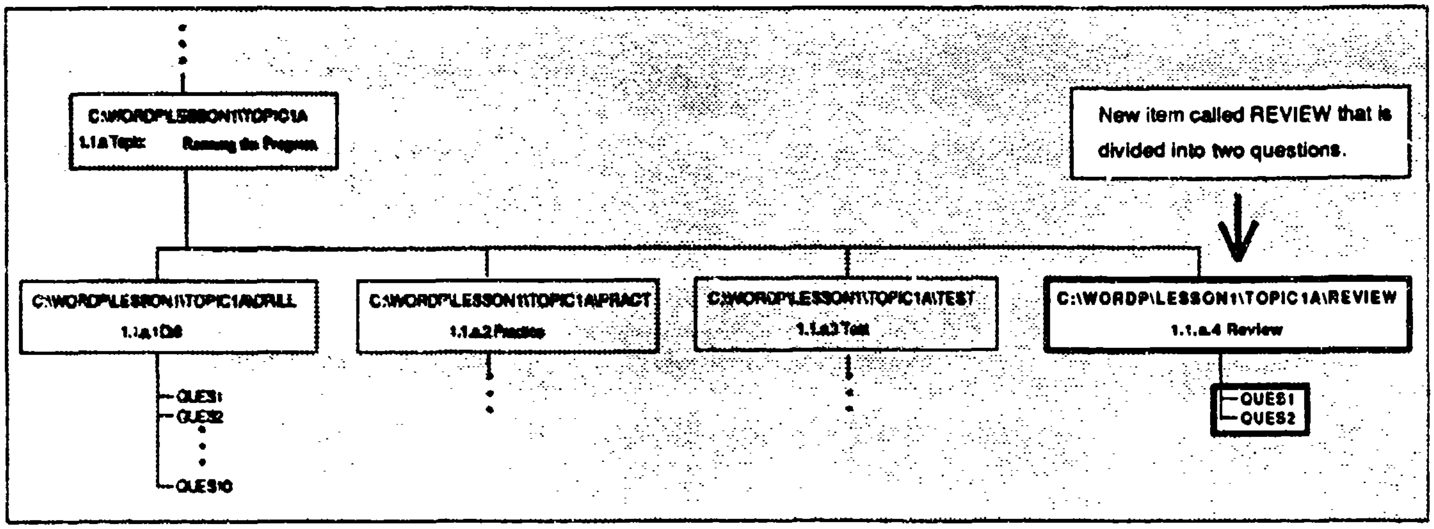

C:\WORDP\LESSON1\TOPIC1A
1.1.a Topic: Running the Program
C:\WORDP\LESSON1\TOPIC1A\DRILL
1.1.a.1 Drill
QUES1
QUES2
QUES10
C:\WORDP\LESSON1\TOPIC1A\PRACT
1.1.a.2 Practice
C:\WORDP\LESSON1\TOPIC1A\TEST
1.1.a.3 Test
New item called REVIEW that is divided into two questions.
C:\WORDP\LESSON1\TOPIC1A\REVIEW
1.1.a.4 Review
QUES1
QUES2

Figure 18. Adding a directory to the courseware directory tree.

Table 7

Adding a Directory to the Courseware Directory Tree

Procedure for Adding a Directory to the Courseware Directory Tree	
1. Add the new directory to the appropriate part of the directory tree.	• Move to the directory that will contain the new subdirectory. • Create the new subdirectory.
2. Create the storyboard, graphic, and animation files.	• Move to the new directory. • Create the new storyboard, graphic, and animation files.
3. Build segment files for the new storyboard files	• Use `CATSC` to build the segments for each new storyboard file. • Watch for any compilation errors. If there are such errors, reedit the appropriate storyboard files to fix the errors, then go back to the beginning of step 3 to rebuild the segments for those storyboard files.
4. Test the new segment files.	• Use `CATSVIEW` to test the new segment files and to verify that the new graphic and animation files are correct.
5 Create a `BUILD.BAT` and `QBUILD.BAT` file for the new directory.	• Create a `BUILD.BAT` for the new directory. The easiest way to create a new batch file is to alter a copy of an existing batch file, preferably one that belongs to a similar directory. Find an existing `BUILD.BAT` file and copy it into the new directory. • Edit the new `BUILD.BAT` file and revise its DOS commands to build segments for the new storyboard files and to build the package segment for the new directory. Save the `BUILD.BAT` file as a nondocument or ASCII file. • Create a `QBUILD.BAT` for the new directory. Again, the easiest way to create this new batch file is to alter a copy of a similar one. • Edit the `QBUILD.BAT` file with your word processor and revise its DOS commands to build the package segment for the new directory. Save the `QBUILD.BAT` file as a nondocument or ASCII file.
6. If the courseware in the new directory must be distributed, create a `DISTRIB.BAT` file for the directory.	• Try to use a copy of an existing `DISTRIB.BAT` file as your starting point. For more information about creating a `DISTRIB.BAT` file, see the instructions for making distributable courseware in Table 2.
7. Test all the new batch files.	• Execute the new batch files to verify that they are correct and to verify that the new directory's package segment is correct.
8. Locate any existing batch files that are affected.	• Determine which batch files should be updated. (Step 8 focuses on locating the files that are affected by adding a directory to the courseware directory tree. Step 9 covers the changes that should be made to these batch files.) To locate the affected batch files: a. Move up one directory. The `BUILD.BAT`, `QBUILD.BAT`, and `DISTRIB.BAT` files must be updated. b. Move up one directory. The `BUILD.BAT` and `DISTRIB.BAT` files in this directory must be updated. Continue this step until you reach the topmost directory in the courseware directory tree. Steps a and b define a path of directories beginning with the directory directly above the new directory and continuing to the topmost directory in the CBI directory tree.

Table 7 (Continued)

Procedure for adding a Directory to the Courseware Directory Tree	
9. Update the affected batch files.	• To update the batch files, you must know how batch files work. Appendix A describes in detail how `BUILD.BAT` and `QBUILD.BAT` work. Appendix B describes in detail how `DISTRIB.BAT` works. • Once you have added a new directory, you must add DOS commands that reference the new directory's package segment and any graphic or animation files in the new directory. For each `BUILD.BAT` that is affected you should: • Add commands to build segments for the new storyboard files. • Add a command to build a package segment for the new directory. • Revise the command that builds the package segment for the directory directly above the new directory to include the package segment for the new directory. For the `QBUILD.BAT` that is affected, you should: • Revise the command that builds the package segment for the directory directly above the new directory to include the package segment for the new directory. For each `DISTRIB.BAT` that is affected you should: • Add commands that copy any new graphic or animation files to the `INSTALL` directory.
10. Test all the updated batch files.	• Execute each updated batch file to verify that it is correct. • Executing each updated batch will also give you the opportunity to verify that the package segment for the new directory does not conflict with the rest of the courseware in the directory tree.

Table 8

Removing a Directory from the Courseware Directory Tree

Procedure for Removing a Directory from the Courseware Directory Tree	
1. Remove the directory.	• Move to the directory that needs to be removed. • Remove all files in the directory. • Move up to the directory immediately above the directory you want to remove. • Remove the directory.
2. Locate any existing batch files that are affected.	• Determine which batch files should be updated. (Step 2 focuses on locating the batch files that are affected by removing a directory from the courseware directory tree. Step 3 covers the changes that should be made to these batch files.) To locate the affected batch files: a. Update the `BUILD.BAT, QBUILD.BAT,` and `DISTRIB.BAT` files in the current directory. b. Move up one directory. Any `BUILD.BAT` and `DISTRIB.BAT` files in this directory must be updated. Continue this step until you reach the topmost directory in the CBI directory tree. Steps a and b define a path of directories beginning with the directory directly above the removed directory and continuing to the topmost directory in the CBI directory tree. All batch files in the directory directly above the removed directory must be updated. In addition, any `BUILD.BAT` and `DISTRIB.BAT` file in the path must be updated.
3. Update the affected batch files.	• To update the batch files, you must know how batch files work. Appendix A describes how `BUILD.BAT` and `QBUILD.BAT` work. Appendix B describes how `DISTRIB.BAT` works. Once you have removed a directory, remove the DOS commands that reference the directory's package segment and any graphic or animation files that were in the directory. For the `BUILD.BAT` that is affected you should: • Remove all references to the eliminated files and directories. For the `QBUILD.BAT` that is affected you should: • Remove all references to the eliminated files and directories. For each `DISTRIB.BAT` that is affected you should: • Remove all references to the eliminated files and directories.
4. Test all the updated files.	• Execute each updated batch file to verify that it is correct. • Executing each updated batch file will also give you the opportunity to verify that removing the directory does not conflict with the rest of the courseware in the directory tree.

GLOSSARY

To avoid ambiguity, the key terms used in the CATS maintenance process are defined below. These definitions have been adapted specifically for this document. They may therefore be more restrictive than, or different from, commonly accep?•d definitions for these terms.

ASCII

American Standard Code for Information Interchange. A widely accepted standard for encoding alphanumeric data.

Animation file

A file containing an animation sequence. Animation files are created by courseware authors using special programs called animation editors such as "Show Partner" and "Grasp."

Authoring tool(s) or authoring system

A software program (or programs) designed to produce and maintain CBI, e.g., CATS.

Batch file

A text file that contains a sequence of DOS commands. When the batch file is executed, its DOS commands are performed in the order that they appear in the batch file.

BUILD.BAT

A DOS batch file that resides in every directory within the courseware directory tree. This batch file is designed to build the segment file for the directory and all of its subdirectories.

CATS

CBI Authoring Tools System. A Government-owned authoring system. CATS includes: (1) A program named CATSC that converts storyboards into segments, (2) a program named CATSM that binds segments, and (3) a program named CATSVIEW that displays segments, graphic files, and animation files as CBI.

CATSC

An acronym for the CATS compiler; a program in CATS that compresses and encodes ASCII storyboards into segments.

CATSM

An acronym for the CATS multiple segment linker; a program in CATS that binds two or more segments.

CATSVIEW

A program in CATS that displays segments, graphic files, and animation files as CBI.

CBI

Computer-based instruction. Instruction or training delivered on a computer. Sometimes referred to as computer-assisted learning (CAL) or computer-assisted instruction (CAI).

Courseware author

A person who creates CATS courseware.

Courseware directory tree

A directory structure used to organize CATS courseware files.

Courseware files

The computer files used to create and present CBI with CATS. There are two types of courseware files: (1) source files and (2) presentation files.

DISTRIB.BAT

A DOS batch file that resides in a directory that represents a distributable portion of the courseware within the courseware directory tree. This batch file is designed to build the presentation disks for the courseware in the directory.

Distributable courseware

Portions of the courseware in a courseware directory tree that can be separately and autonomously placed on floppy disks and distributed to students. A courseware author designates the distributable courseware of a courseware directory tree as part of the courseware design process. A directory in the courseware directory tree that represents distributable courseware will always contain a DISTRIB.BAT file. This is the batch file that builds presentation disks for distributable courseware.

Graphic file

A file containing a graphic image. Graphic files are created by courseware authors using special programs called graphic editors. "Dr. Halo," produced by Media Cybernetics, and "PC-Paintbrush," produced by Z-soft Corporation, are two examples of graphic editors that produce graphic files.

Instructional outline

An organization of courseware material that divides information into items and subitems. Items in an instructional outline can be topics of information such as "Basics of Word Processing" or "U.S. History," or instructional tools such as tests, drills, lessons, etc.

Presentation disk

A presentation disk contains the presentation files, or a subset of the presentation files, used to present CBI. Presentation disks contain the CATS presentation software, lesson segment files, profile files, graphic files, and animation files. One of the presentation disks contains only the profile files. This disk is called the student disk.

Presentation files

All of the computer files required to present CBI. Includes CATS presentation software, lesson segment files, profile files, DIR files, graphic files, and animation files. (Note: Graphic files and animation files are also source files.)

Presentation software

A software program that presents CBI. For example, the CATS presentation program for your lesson is called `EGA.EXE`.

`QBUILD.BAT`

A DOS batch file that resides in every directory within the courseware directory tree. This batch file is designed to build the segment file for the directory using the existing segment files in its directory. (Note: Because this batch file is designed to run quickly, it will build only the segment file for its directory. Unlike `BUILD.BAT`, it will not build segment files for any subdirectories.)

Segment file (or Segment)

A presentation file produced by `CATSC` that is an encoded and compressed version of a storyboard file or a collection of storyboard files.

Source disk

A source disk contains the source files for a CBI lesson, or a subset of the source files since the source files together may exceed the capacity of a single disk.

Source files

Source files are computer files used to create and maintain CBI using CATS. Storyboard files, graphic files, animation files, batch files, and CATS variable files are source files. (Note: Graphic and animation files are also presentation files.)

Storyboard

A description of one screen and/or one operator interaction in CATS courseware. Storyboards are stored in storyboard files.

Storyboard file

An ASCII file that contains one or more storyboards and is made using a text editor or word processing program. Storyboard files are source files and contain the basic building blocks of the lesson. They contain information that controls what the student sees on the screen, how the CBI will react to student responses, what data are collected by the CBI, and the lesson support files used by the CBI.

Student disk

A student disk is a presentation disk that contains only the profile files (i.e., `PROFILE.CAT` and `PROFILE.FIX` files). Each student needs his or her own copy of this disk to run each lesson. A student disk is used as a repository for individual student data. The `PROFILE.CAT` file on an individual student disk will be updated automatically to reflect the student's progress during the student's use of the lesson.

APPENDIX A

HOW BUILD.BAT AND QBUILD.BAT WORK

HOW BUILD.BAT AND QBUILD.BAT WORK

BUILD.BAT and QBUILD.BAT are batch files that build segment files for a directory. Courseware authors create these files for every directory in the courseware directory tree. Batch files make it easy to build segment files. By using batch files, the courseware developer only has to type one command (either BUILD.BAT or QBUILD.BAT) instead of remembering a whole series of commands.

BUILD.BAT

The objective of the BUILD.BAT file is to build the segment file for its directory from scratch. To do this, BUILD.BAT performs the following steps:

1. Removes all segment files from the current directory and any subdirectories of the current directory.

2. Builds new segment files for all the storyboard files in the subdirectory.

3. Builds a new package segment for the subdirectory.

4. Builds a new package segment for the current directory.

Each of these steps is accomplished by one or more DOS commands in the BUILD.BAT file. When the batch file is executed, the DOS commands are executed in the order that they appear in the batch file.

All BUILD.BAT batch files have the same objectives, and they use the same steps to reach their objectives. However, since BUILD.BAT files reside in different parts of the directory tree, they will use slightly different DOS commands to accomplish these steps. To describe in detail how BUILD.BAT files work, we will discuss the DOS commands used by one EUILD.BAT file in the word processing example. Keep in mind the steps that all BUILD.BAT files must accomplish while we discuss the specific DOS commands in the example.

Figure A-1 illustrates a portion of the courseware directory tree for the word processing CBI. We will use the BUILD.BAT file in the TOPIC1A directory as our example.

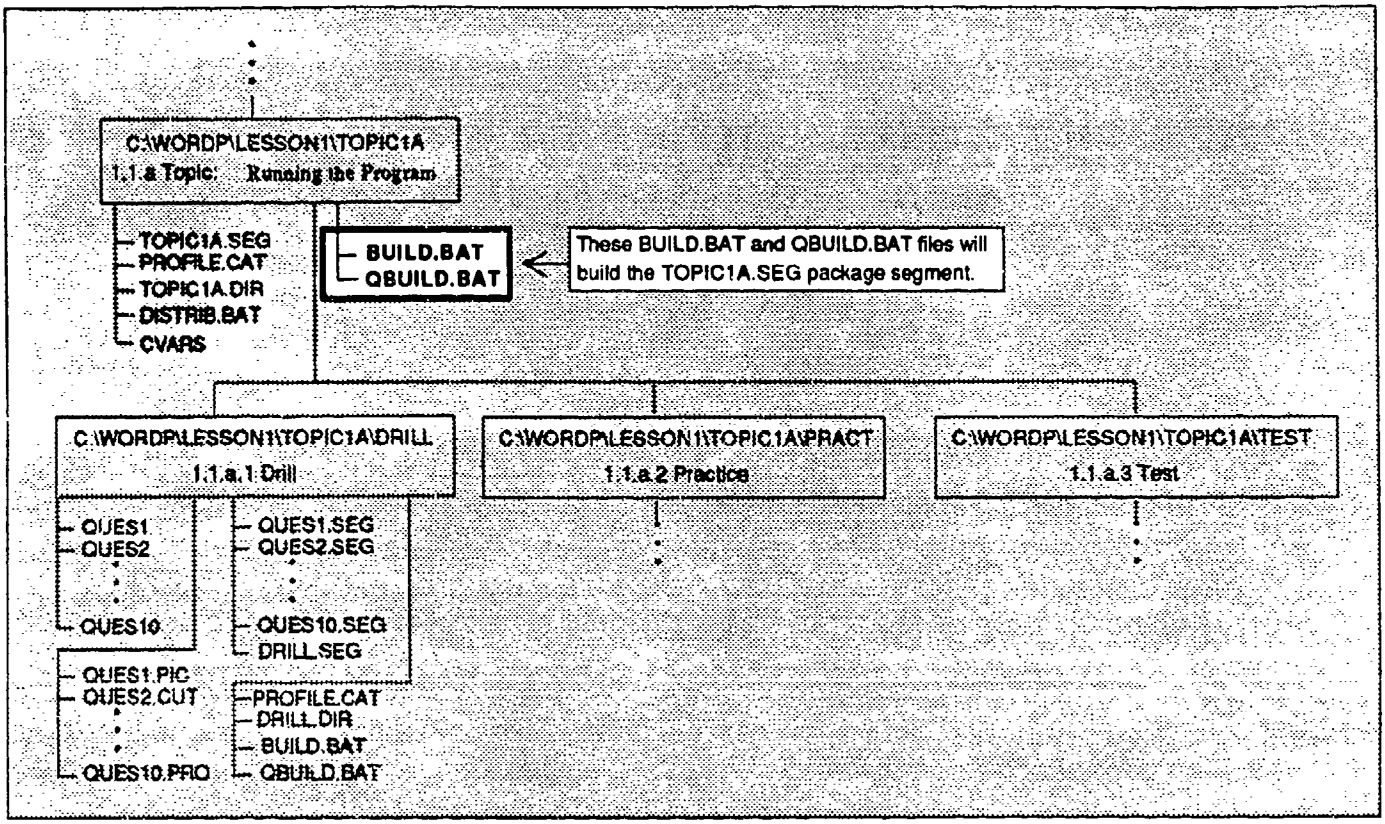
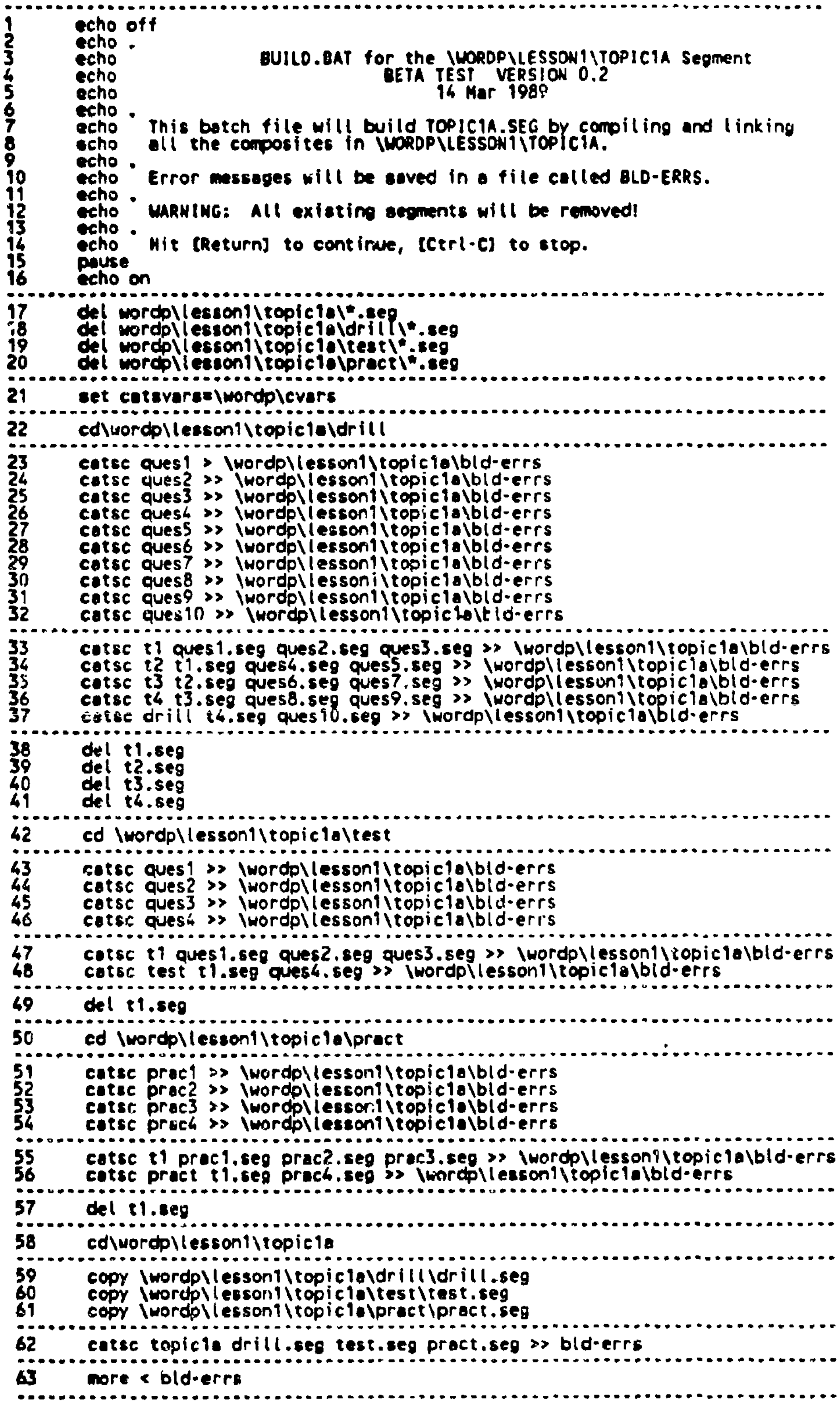


Figure A-1. The `BUILD.BAT` and `QBUILD.BAT` files for the `TOPIC1A` directory will build the package segment called `TOPIC1A.SEG`.

Figure A-2 shows the contents of this `BUILD.BAT` file. The numbers to the left of each line in the `BUILD.BAT` file are called "line numbers." These line numbers are referenced below in describing what the DOS commands do.

Lines 1 - 16 Presents information to the user telling what the batch file will do. Gives the user the option of quitting or continuing. (The echo features allow the user to see or not to see what commands are being executed in the batch file.)

Lines 17 - 20 Removes all the segment files that this batch file is about to rebuild. If these segment files were not removed and the batch file were stopped before completion, the user might then accidentally use an old version of a segment file.



```
--------------------------------------------------------------------------------
echo off
echo .
echo             BUILD.BAT for the \WORDP\LESSON1\TOPIC1A Segment
echo                            BETA TEST  VERSION 0.2
echo                                 14 Mar 1989
echo .
echo   This batch file will build TOPIC1A.SEG by compiling and linking
echo   all the composites in \WORDP\LESSON1\TOPIC1A.
echo .
echo   Error messages will be saved in a file called BLD-ERRS.
echo .
echo   WARNING:  All existing segments will be removed!
echo .
echo   Hit [Return] to continue, [Ctrl-C] to stop.
pause
echo on
--------------------------------------------------------------------------------
del wordp\lesson1\topic1a\*.seg
del wordp\lesson1\topic1a\drill\*.seg
del wordp\lesson1\topic1a\test\*.seg
del wordp\lesson1\topic1a\pract\*.seg
--------------------------------------------------------------------------------
set catsvars=\wordp\cvars
--------------------------------------------------------------------------------
cd\wordp\lesson1\topic1a\drill
--------------------------------------------------------------------------------
catsc ques1 > \wordp\lesson1\topic1a\bld-errs
catsc ques2 >> \wordp\lesson1\topic1a\bld-errs
catsc ques3 >> \wordp\lesson1\topic1a\bld-errs
catsc ques4 >> \wordp\lesson1\topic1a\bld-errs
catsc ques5 >> \wordp\lesson1\topic1a\bld-errs
catsc ques6 >> \wordp\lesson1\topic1a\bld-errs
catsc ques7 >> \wordp\lesson1\topic1a\bld-errs
catsc ques8 >> \wordp\lesson1\topic1a\bld-errs
catsc ques9 >> \wordp\lesson1\topic1a\bld-errs
catsc ques10 >> \wordp\lesson1\topic1a\bld-errs
--------------------------------------------------------------------------------
catsc t1 ques1.seg ques2.seg ques3.seg >> \wordp\lesson1\topic1a\bld-errs
catsc t2 t1.seg ques4.seg ques5.seg >> \wordp\lesson1\topic1a\bld-errs
catsc t3 t2.seg ques6.seg ques7.seg >> \wordp\lesson1\topic1a\bld-errs
catsc t4 t3.seg ques8.seg ques9.seg >> \wordp\lesson1\topic1a\bld-errs
catsc drill t4.seg ques10.seg >> \wordp\lesson1\topic1a\bld-errs
--------------------------------------------------------------------------------
del t1.seg
del t2.seg
del t3.seg
del t4.seg
--------------------------------------------------------------------------------
cd \wordp\lesson1\topic1a\test
--------------------------------------------------------------------------------
catsc ques1 >> \wordp\lesson1\topic1a\bld-errs
catsc ques2 >> \wordp\lesson1\topic1a\bld-errs
catsc ques3 >> \wordp\lesson1\topic1a\bld-errs
catsc ques4 >> \wordp\lesson1\topic1a\bld-errs
--------------------------------------------------------------------------------
catsc t1 ques1.seg ques2.seg ques3.seg >> \wordp\lesson1\topic1a\bld-errs
catsc test t1.seg ques4.seg >> \wordp\lesson1\topic1a\bld-errs
--------------------------------------------------------------------------------
del t1.seg
--------------------------------------------------------------------------------
cd \wordp\lesson1\topic1a\pract
--------------------------------------------------------------------------------
catsc prac1 >> \wordp\lesson1\topic1a\bld-errs
catsc prac2 >> \wordp\lesson1\topic1a\bld-errs
catsc prac3 >> \wordp\lesson1\topic1a\bld-errs
catsc prac4 >> \wordp\lesson1\topic1a\bld-errs
--------------------------------------------------------------------------------
catsc t1 prac1.seg prac2.seg prac3.seg >> \wordp\lesson1\topic1a\bld-errs
catsc pract t1.seg prac4.seg >> \wordp\lesson1\topic1a\bld-errs
--------------------------------------------------------------------------------
del t1.seg
--------------------------------------------------------------------------------
cd\wordp\lesson1\topic1a
--------------------------------------------------------------------------------
copy \wordp\lesson1\topic1a\drill\drill.seg
copy \wordp\lesson1\topic1a\test\test.seg
copy \wordp\lesson1\topic1a\pract\pract.seg
--------------------------------------------------------------------------------
catsc topic1a drill.seg test.seg pract.seg >> bld-errs
--------------------------------------------------------------------------------
more < bld-errs
--------------------------------------------------------------------------------
```

Figure A-2. **BUILD.BAT files contain DOS commands to build a directory's segment file from scratch.**

Line 21 Sets the CATS variable file location as WORDP\CVARS. This enables CATSC to locate the CATS author variables for the word processing CBI.

Line 22 Moves to a bottom-level subdirectory called DRILL. BUILD.BAT starts at the lowest level of subdirectories and proceeds to build segment files from the "bottom up." This is because segment files in the upper subdirectories are packages of the segments from their lower subdirectories.

Lines 23 - 32 Builds segment files for each storyboard file in the DRILL subdirectory. Any compilation errors will be copied to a file called BLD-ERRS in the TOPIC1A directory. (The ">" and ">>" are parts of DOS commands that cause messages from a program to be sent to a text file.)

Lines 33 - 37 Builds a package segment DRILL.SEG for the DRILL subdirectory. Any compilation errors will be copied to a file called BLD-ERRS in the TOPIC1A directory.

Lines 38 - 41 Removes the temporary segment files that were created during the process of building the DRILL.SEG package segment. (DOS commands must fit on only one line. Therefore, CATSC commands are sometimes broken up and the intermediate segments are placed in temporary files.)

Line 42 Moves to another bottom-level subdirectory called TEST.

Lines 43 - 46 Builds segment files for each storyboard file in the TEST subdirectory. Any compilation errors will be added to the BLD-ERRS file in the TOPIC1A directory.

Lines 47 - 48 Builds a package segment TEST.SEG for the TEST subdirectory. Any compilation errors will be copied to the BLD-ERRS file in the TOPIC1A directory.

Line 49 Removes the temporary segment file that was created during the process of building the TEST.SEG package segment.

Line 50 Moves to the last bottom-level subdirectory called PRACT.

Lines 51 - 54 Builds segment files for each storyboard file in the PRACT subdirectory. Any compilation errors will be copied to the BLD-ERRS file in the TOPIC1A directory.

Lines 55 - 56 Builds a package segment PRACT.SEG for the PRACT subdirectory. Any compilation errors will be copied to the BLD-ERRS file in the TOPIC1A directory.

Line 57 Removes the temporary segment file that was created during the process of building the PRACT.SEG package segment.

Line 58 Moves to the next highest level subdirectory TOPIC1A.

Lines 59 - 61 Copies the package segments DRILL.SEG, TEST.SEG, and PRACT.SEG from the lower-level subdirectories.

Line 62 Builds the package segments DRILL.SEG, TEST.SEG, and PRACT.SEG into a package segment called TOPIC1A.SEG. Any compilation errors will be copied to the BLD-ERRS file in the TOPIC1A directory. (Since this command is being executed in the TOPIC1A directory, it did not have to provide a full path name for the BLD-ERRS file.

Line 63 Displays any compilation errors on the screen (via the DOS MORE command).

QBUILD.BAT

The objective of the QBUILD.BAT file is to quickly build the segment file for its directory without rebuilding any of the segments for storyboard files or subdirectories. To do this, the QBUILD.BAT performs the following steps:

1. Removes the package segment file from the current directory.

2. Builds a new package segment for the current directory.

Each of these steps is accomplished by one or more DOS commands in the QBUILD.BAT file. When the batch file is executed, the DOS commands are executed in the order in which they appear in the batch file.

Like BUILD.BAT files, all QBUILD.BAT batch files use slightly different DOS commands to reach the same objective. The following example describes in detail how QBUILD.BAT files work, discussing the DOS commands used by one QBUILD.BAT file in the word processing example. Again, keep in mind the objective of all QBUILD.BAT files while looking at the specific DOS commands in the example.

We used the BUILD.BAT file from the TOPIC1A directory to discuss how BUILD.BAT files work. Now we will use the QBUILD.BAT file from the same directory as our example for QBUILD.BAT files. Figure A-3 shows the contents of this QBUILD.BAT file.

```
1   echo off
2   echo .
3   echo        QBUILD.BAT for the \WORDP\LESSON1\TOPIC1A Segment
4   echo               BETA TEST  VERSION 0.2
5   echo                  14 Mar 1989
6   echo .
7   echo   This batch file will build TOPIC1A.SEG.
8   echo .
9   echo   Please make sure that all the storyboard files have been compiled
10  echo   BEFORE using this batch file to build TOPIC1A.SEG.
11  echo .
12  echo   You can either compile the storyboard files individually, or you
13  echo   can use the batch file named BUILD.BAT to compile them all
14  echo   at once.
15  echo .
16  echo   Error messages will be saved in a file called BLD-ERRS.
17  echo .
18  echo   WARNING: Any existing TOPIC1A.SEG will be removed!
19  echo .
20  echo   Hit [Return] to continue, [Ctrl-C] to stop.
21  pause
22  echo on
23  del wordp\lesson1\topic1a\*.seg
24  set catsvars=\wordp\cvars
25  copy \wordp\lesson1\topic1a\drill\drill.seg
26  copy \wordp\lesson1\topic1a\test\test.seg
27  copy \wordp\lesson1\topic1a\pract\pract.seg
28   catsc topic1a drill.seg test.seg pract.seg > bld-errs
29   more < bld-errs
```

Figure A-3. QBUILD.BAT files contain DOS commands to quickly build a directory's segment file.

Lines 1 - 22	Presents information to the user telling what the batch file will do. Gives the user the option of quitting or continuing.
Line 23	Removes the package segment file from the current directory. If this segment file were not removed and the batch file were stopped before completion, the user might then accidentally use an old version of a segment file.
Line 24	Sets the CATS variable file location as WORDP\CVARS. This enables CATSC to locate the CATS author variables for the word processing CBI.
Lines 25 - 27	Copies the package segments DRILL.SEG, TEST.SEG, and PRACT.SEG from the lower-level subdirectories.
Line 28	Builds a package segment TOPIC1A.SEG for the TOPIC1A subdirectory.
Line 29	Displays any compilation errors on the screen (via the DOS MORE command).

APPENDIX B

HOW DISTRIB.BAT WORKS

HOW DISTRIB.BAT WORKS

The objective of the DISTRIB.BAT file is to build presentation disks for its directory. To do this the DISTRIB.BAT performs the following steps:

1. Creates a temporary directory called INSTALL that will contain all the files that need to be placed on the presentation disks.

2. Copies the presentation program and execution batch files to the INSTALL directory.

3. Copies all the graphic and animation files used by the package segment to the INSTALL directory.

4. Copies the package segment and the profile file for the current directory to the INSTALL directory.

5. Executes a program called BLD_INST.EXE that will calculate how many presentation disks are needed, format the presentation disks, then copy all the files from the INSTALL directory to the appropriate presentation disks.

6. Removes all the files in the INSTALL directory, then removes the INSTALL directory.

Each of these steps is accomplished by one or more DOS commands in the DISTRIB.BAT file. When the batch file is executed, the DOS commands are executed in the order in which they appear in the batch file.

All DISTRIB.BAT batch files have the same objective and use the same steps to reach their objective. However, since DISTRIB.BAT files reside in different parts of the directory tree, they use slightly different DOS commands to accomplish these steps. To describe in detail how DISTRIB.BAT files work, the following pages discuss the DOS commands used by one DISTRIB.BAT file in the word processing example. During this discussion of the specific DOS commands used by the example DISTRIB.BAT, keep in mind the steps that all DISTRIB.BAT files must accomplish.

Figure B-1 illustrates the location of a distributable portion of the courseware directory tree for the word processing CBI. Since topic 1A of lesson 1 is distributable, its directory contains the DISTRIB.BAT file. Figure B-2 shows the contents of this DISTRIB.BAT file. The numbers shown in the figure for each line in the DISTRIB.BAT file are called "line numbers." These line numbers are used below to isolate commands in the DISTRIB.BAT file and to describe what these commands do.

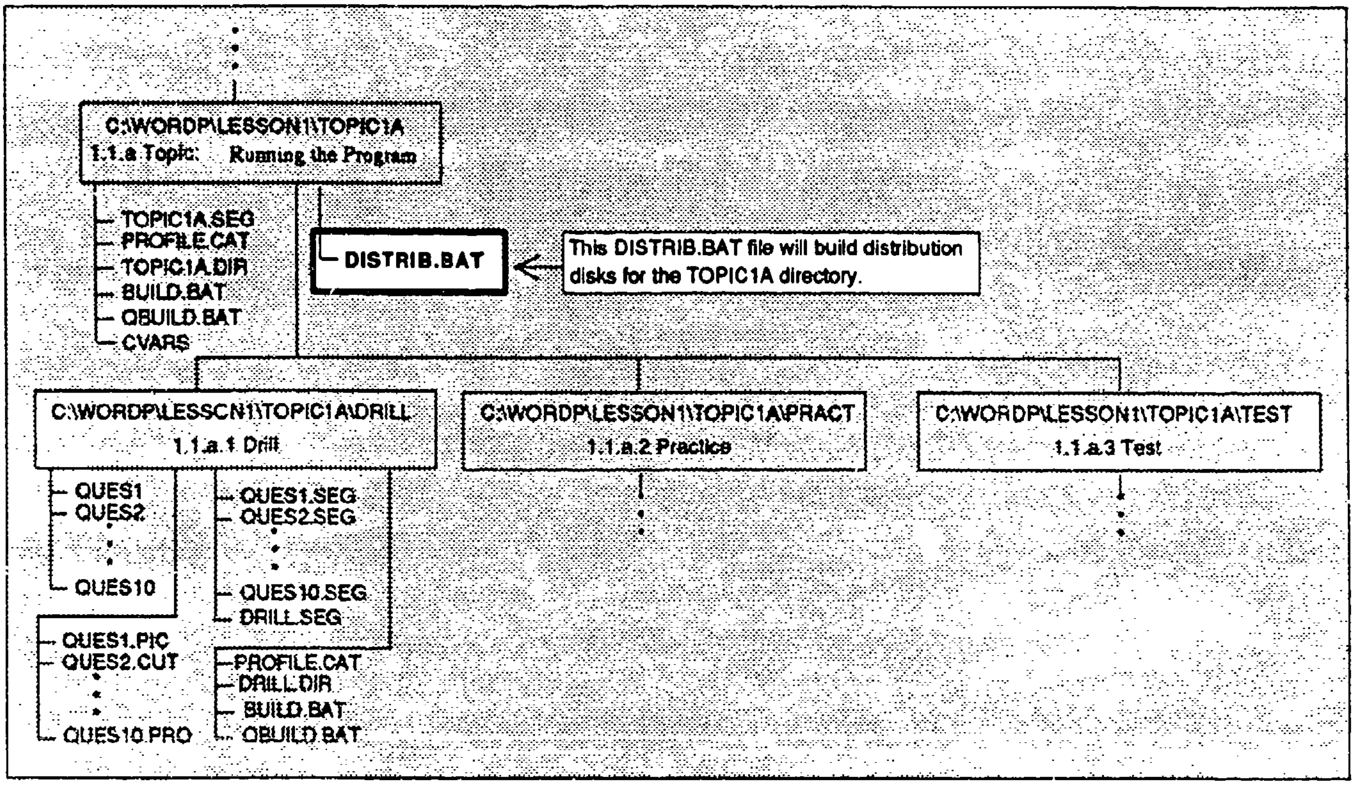


Figure B-1. The `DISTRIB.BAT` file for the `TOPIC1A` directory that will build its presentation disks.

Lines 1 - 18 Presents information to the user telling what the batch file will do. Gives the user the option of quittirg or continuing. (The echo features allow the user to see or not see what commands are being executed.)

Line 19 Makes a temporary directory called `INSTALL`. This directory will be used as a repository for all the files that must be placed on the presentation disks for the `TOPIC1A` directory.

Lines 20 - 21 Copies the program `EGA.EXE` that presents CBI to the students and the batch file `RUN_LES.BAT` that the students use to execute the CBI to the `INSTALL` directory.

Lines 22 - 24 Copies all the graphic and animation files from the `DRILL` directory to the `INSTALL` directory.

Lines 25 - 27 Copies all the graphic and animation files from the `TEST` directory to the `INSTALL` directory.

Lines 28 - 30 Copies all the graphic and animation files from the PRACT directory to the INSTALL directory.

Line 31 - 32 Copies the TOPIC1A.SEG segment file and the PROFILE.CAT file that identifies the segment to the INSTALL directory.

Line 33 Moves to the INSTALL directory to begin building presentation disks.

Lines 34 - 45 Presents information to the user saying that the generic student disk is about to be built. Asks the user to place the generic student disk in the "a:" drive.

Lines 46 - 48 Formats the generic student disk.

Lines 49 - 51 Places the PROFILE.CAT file that identifies the TOPIC1A.SEG segment onto the generic student disk. Makes backup copy of PROFILE.CAT and names it PROFILE.FIX. Removes the PROFILE.CAT file so that it will not be placed on the rest of the presentation disks.

Lines 52- 60 Presents information to the user saying that the rest of the presentation disks (which are installation disks) are about to be built.

Line 61 Executes the program called BLD_INST.EXE. This program will build the rest of the presentation disks. To do this, it will first determine how many presentation disks are needed, given the number of files in the INSTALL directory. Then, for each presentation disk, it will ask the user to place the disk in the "a:" drive, format the disk, and copy the appropriate files to the disk.

Lines 62- 70 Moves to the TOPIC1A directory, removes all the files in the INSTALL directory, then removes the INSTALL directory.

```
echo off
echo .
echo            Build Installation Disks and Generic Student Disk
echo                      BETA TEST VERSION 0.25
echo                           14 Mar 1989
echo .
echo .
echo  This batch file will build installation disks and a generic
echo  student disk for the TOPIC1A directory.
echo .
echo .
echo  WARNING:  Be sure that you have build up-to-date segment files
echo            for the directory before you use this batch file!
echo .
echo  Hit [Return] to continue, [Ctrl-C] to stop.
echo .
pause
echo on

mkdir \wordp\lesson1\topic1a\install

copy \cats\bin\ega.exe \wordp\lesson1\topic1a\install
copy \cats\bin\run_les.bat \wordp\lesson1\topic1a\install\go.bat

copy \wordp\lesson1\topic1a\drill\*.pic \wordp\lesson1\topic1a\install
copy \wordp\lesson1\topic1a\drill\*.cut \wordp\lesson1\topic1a\install
copy \wordp\lesson1\topic1a\drill\*.pro \wordp\lesson1\topic1a\install

copy \wordp\lesson1\topic1a\test\*.pic \wordp\lesson1\topic1a\install
copy \wordp\lesson1\topic1a\test\*.cut \wordp\lesson1\topic1a\install
copy \wordp\lesson1\topic1a\test\*.pro \wordp\lesson1\topic1a\install

copy \wordp\lesson1\topic1a\pract\*.pic \wordp\lesson1\topic1a\install
copy \wordp\lesson1\topic1a\pract\*.cut \wordp\lesson1\topic1a\install
copy \wordp\lesson1\topic1a\pract\*.pro \wordp\lesson1\topic1a\install

copy \wordp\lesson1\topic1a\topic1a.seg \wordp\lesson1\topic1a\install
copy \wordp\lesson1\topic1a\profile.cat \wordp\lesson1\topic1a\install

cd \wordp\lesson1\topic1a\install

echo off
echo .
echo .
echo .
echo     Building a generic student disk for the TOPIC1A CBI.
echo .
echo .
echo .
echo    Please place the generic student disk in drive A:
echo .
pause
echo on

echo . > ret
echo n >> ret
format a: < ret
```

Figure B-2. Listing of the **DISTRIB.BAT** file that contains DOS commands to build presentation disks for the **TOPIC1A** directory.

```
copy profile.cat a:
copy profile.cat a:profile.fix
del profile.cat

echo off
echo .
echo .
echo .
echo     Building installation disks for the TOPIC1A CBI.
echo .
echo .
echo .
echo on

bld_inst.exe a:

cd \wordp\lesson1\topic1a
del \wordp\lesson1\topic1a\install\*
del \wordp\lesson1\topic1a\install\*.seg
del \wordp\lesson1\topic1a\install\*.pic
del \wordp\lesson1\topic1a\install\*.cut
del \wordp\lesson1\topic1a\install\*.pro
del \wordp\lesson1\topic1a\install\*.bat
del \wordp\lesson1\topic1a\install\*.exe
rmdir \wordp\lesson1\topic1a\install
```

Figure B-2. (Continued)

APPENDIX C

HANDLING PRESENTATION DISK SIZE CONSTRAINTS

HANDLING PRESENTATION DISK SIZE CONSTRAINTS

The maintenance system is designed to distribute courseware on low-density floppy disks. Low-density floppy disks can store a maximum of 360 kilobytes, so the program BLD_INST.EXE will not copy a file larger than 360 kilobytes onto a presentation disk. Because of this limitation, no file larger than 360 kilobytes may be distributed. While it is somewhat unlikely, there is a possibility that changes to courseware, such as adding graphic files or adding several storyboards, may cause courseware that once fit on a single floppy disk to expand beyond the 360 kilobyte limit. While it is currently impossible to use the maintenance system to distribute a graphic or animation file that exceeds 360 kilobytes, it is possible to distribute a lesson segment file that exceeds 360 kilobytes.

To distribute a courseware segment that exceeds the 360-kilobyte limit, it is necessary to break up the segment into smaller segment files, link these segment files, and then distribute these multiple linked segments. If the large segment is a "package" of smaller segments, then the smaller segments can be used. However, if the segment resides in a bottom-level directory, it will first be necessary to divide the directory's courseware into subdirectories and then to distribute the new subdirectories' segments.

The "divide and conquer" strategy for making large segments distributable is outlined in Table C-1. If the large segment file is in a bottom-level directory of the courseware directory tree, start at step 1. If the large segment file is not in a bottom-level directory of the courseware directory tree, start at step 8.

To get a practical understanding of the process of distributing an oversized segment file, consider the word processing example (refer to Figure B-1). Suppose that the DRILL segment was over 360 kilobytes. Since the DRILL directory is at the bottom of the courseware directory tree, you would start at step 1 of Table C-1. You would create new subdirectories of the DRILL directory and divide the courseware between the new subdirectories.

Figure C-1 shows that two new subdirectories, QGROUP1 and QGROUP2, have been added to the DRILL directory and that the courseware has been divided between the two new directories.

Figure C-2 shows the contents of the DISTRIB.BAT file for the new courseware file organization. The numbers by each line in the new DISTRIB.BAT file are called "line numbers." As in the example of a normal DISTRIB.BAT file given in Appendix B, line numbers are used to isolate commands in the new DISTRIB.BAT file and to describe what these commands do. This example, however, lists only the line numbers whose commands pertain to distributing multiple linked segments.

Line 28 - 29 Copies the segment files from the QGROUP1 directory and the QGROUP2 directory to the INSTALL directory.

Line 31 Sets the location of the CATS variable file for use by CATSM.

Line 32 Uses CATSM to link the segments from the two subdirectories.

Table C-1

Making Large Segments Distributable

Procedure for Making Large Segments Distributable	
1. Break up the large segment file.	• Use the instructional outline to help decide how to divide the large courseware block into smaller sections. From the instructional outline, design an extension of subdirectories under the current directory that will house these smaller sections. • Use the DOS mkdir command to add the new subdirectories in the directory that houses the overly large segment. • Break up the courseware for the overly large segment. If an individual storyboard file is producing an overly large segment, use your word processor to divide the file into smaller files, then place these smaller files in the new directories. If the overly large segment is the product of several storyboard files, move these files into the new directories.
2. Build segment files for any new storyboard files.	• Use CATSC to build the segments for each new storyboard file. • Watch for any compilation errors. If there are such errors, reedit the appropriate storyboard files to fix the error, then go back to the beginning of step 2 to rebuild the segments for those storyboard files.
3. Test the new segment file.	• Use CATSVIEW to test the new segment files.
4. Create a BUILD.BAT and QBUILD.BAT file for each new directory.	• Create a BUILD.BAT for each new directory. The easiest way to create a new batch file is to alter a copy of an existing batch file, preferably one that belongs to a similar directory. Therefore, find an existing BUILD.BAT file and copy it into the new directory. (See Appendix A for more details.) • Edit each new BUILD.BAT file with your word processor and revise its DOS commands to build segments for the new storyboard files and to build the segment for the new directory. Save the BUILD.BAT file as a nondocument or ASCII file. • Create a QBUILD.BAT for each new directory. Again, the easiest way to create a new batch file is to alter a copy of a similar one. Therefore, find an existing QBUILD.BAT file and copy it into the new directory. • Edit each new QBUILD.BAT file with your word processor and revise its DOS commands to build the package segment for the new directory. Save the QBUILD.BAT file as a nondocument or ASCII file.
5. Test all the new batch files.	• Execute the new batch files to verify that they are correct and to verify that the new directory's package segment is correct.

Table C-1 (Continued)

6. Locate any existing batch files that are unaffected.	• Determine which batch files should be updated. (Step 6 focuses on locating the batch files that are affected by adding a directory to the courseware directory tree. Step 7 covers the changes tuat should be made to these batch files.) To locate the affected batch files: a. Move up one directory. All batch files in this directory must be updated. b. Move up one directory. Any `BUILD.BAT` and `DISTRIB.BAT` batch files in this directory must be updated. Continue this step until you reach the topmost directory in the courseware directory tree. • Steps a and b define a path of directories beginning with the directory directly above the new directory and continuing to the topmost directory in the CBI directory tree. All batch files in the directory directly above the new directory must be updated. In addition, any `BUILD.BAT` and `DISTRIB.BAT` file in the path must be updated.
7. Update the affected batch files.	• Update the affected batch files. To update the batch files, you must know how batch files work. Appendix A describes in detail how `BUILD.BAT` and `QBUILD.BAT` work. Appendix B describes how `DISTRIB.BAT` works. Once you have added a new directory, you must add DOS commands that reference the new directory's package segment and any graphic or animation files in the new directory. For the `BUILD.BAT` that is affected you should: • Add commands to build segments for the new storyboard files. • Add a command to build a package segment for the new directory. • Revise the command that builds the package segment for the directory directly above the new directory to include the package segment for the new directory. For the `QBUILD.BAT` that is affected you should: • Revise the command that builds the package segment for the directory directly above the new directory to include the package segment for the new directory. For each `DISTRIB.BAT` that is affected you should: • Add commands that copy any new graphic or animation files to the `INSTALL` directory. • Direct the `DISTRIB.BAT` file to distribute courseware using multiple linked segments. (See step 8.)
8. Test all the updated batch files.	• Execute each updated batch file to verify that it is correct. Executing each updated batch file will also give you the opportunity to verify that the package segment for the new directory does not conflict with the rest of the courseware in the directory tree.

Table C-1 (Continued)

9.	Modify `DISTRIB.BAT` to distribute multiple linked segments.	• Modify the `DISTIB.BAT` file for the courseware that you wish to distribute. Two basic changes must be made to the `DISTRIB.BAT` file: • Delete the command that copies the segment file from the current directory to the temporary `INSTALL` directory. • Add commands to copy the segment files from each of the current directory's subdirectories into the temporary `INSTALL` directory. • Add a command to set the location of the CATS variable file (`CVARS`) for use by `CATSM`. • Add a `CATSM` command to link these subdirectory segments within the `INSTALL` directory.

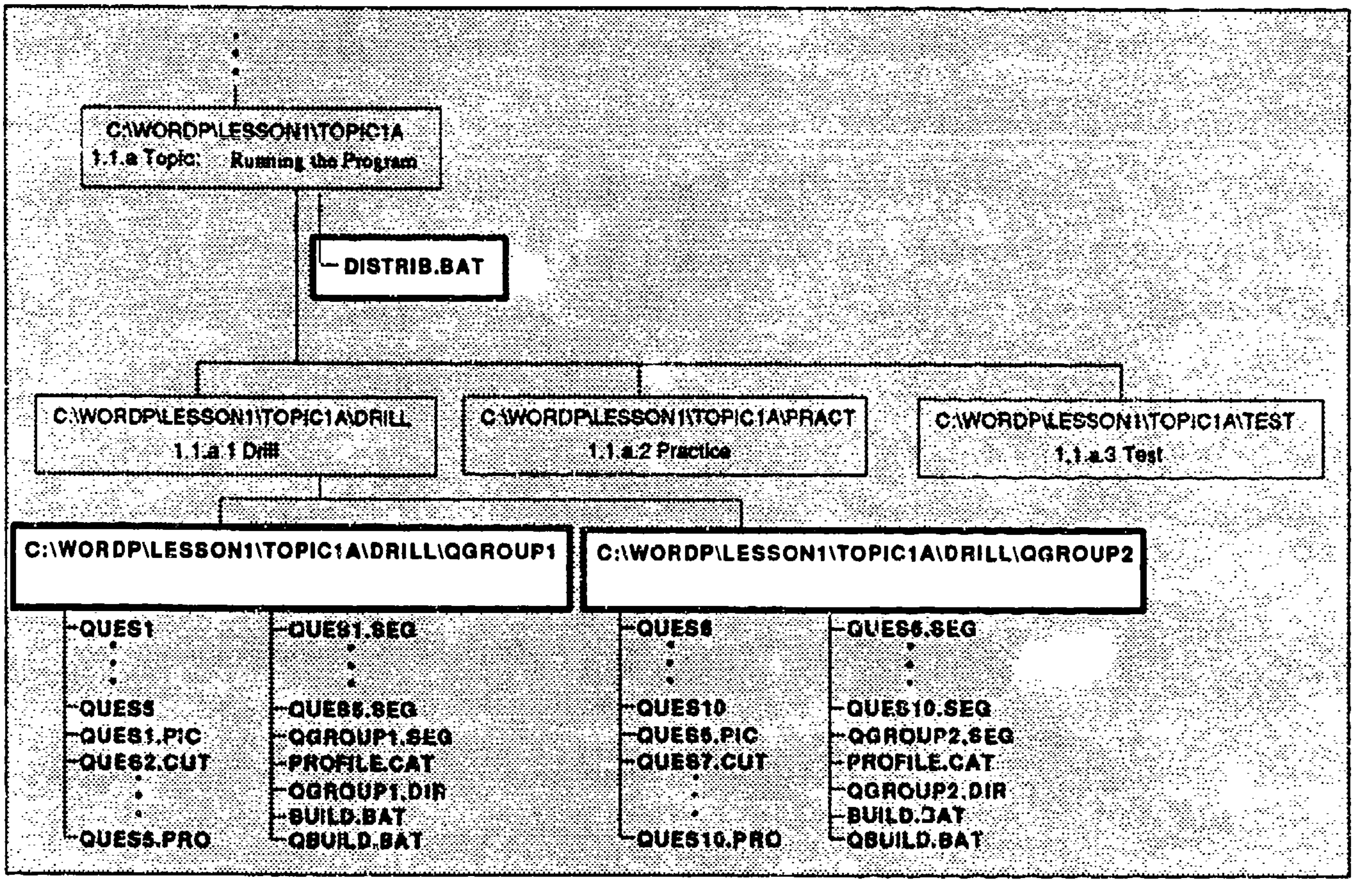


Figure C-1. The `DRILL` directory has been split and its courseware distributed into two new subdirectories.

```
echo off
echo .
echo          Build Installation Disks and Generic Student Disk
echo                   BETA TEST VERSION 0.25
echo                      14 Mar 1989
echo .
echo .
echo  This batch file will build installation disks and a generic
echo  student disk for the TOPIC1A DRILL directory.
echo .
echo .
echo  WARNING: Be sure that you have built up-to-date segment files
echo         for the directory before you use this batch file!
echo .
echo  Hit [Return] to continue, [Ctrl-C] to stop.
echo .
pause
echo on

mkdir \wordp\lesson1\topic1a\drill\install

copy \cats\bin\ega.exe \wordp\lesson1\topic1a\drill\install
copy \cats\bin\run_les.bat \wordp\lesson1\topic1a\drill\install\go.bst

copy \wordp\lesson1\topic1a\drill\qgroup1\*.pic \wordp\lesson1\topic1a\drill\install
copy \wordp\lesson1\topic1a\drill\qgroup1\*.cut \wordp\lesson1\topic1a\drill\install
copy \wordp\lesson1\topic1a\drill\qgroup1\*.pro \wordp\lesson1\topic1a\drill\install

copy \wordp\lesson1\topic1a\drill\qgroup2\*.pic \wordp\lesson1\topic1a\drill\install
copy \wordp\lesson1\topic1a\drill\qgroup2\*.cut \wordp\lesson1\topic1a\drill\install
copy \wordp\lesson1\topic1a\drill\qgroup2\*.pro \wordp\lesson1\topic1a\drill\install

copy \wordp\lesson1\topic1a\drill\qgroup1\qgroup1.seg \wordp\lesson1\topic1a\drill\install
copy \wordp\lesson1\topic1a\drill\qgroup2\qgroup2.seg \wordp\lesson1\topic1a\drill\install

cd \wordp\lesson1\topic1a\drill\install

set catsvars=\wordp\cvars

catsm qgroup1.seg qgroup2.seg > \wordp\lesson1\topic1a\drill\bld-errs

echo off
echo .
echo .
echo    Building a generic student disk for the TOPIC1A CBI.
echo .
echo .
echo .
echo  Please place the generic student disk in drive A:
echo .
pause
echo on

echo . > ret
echo n >> ret
format a: < ret

copy profile.cat a:
copy profile.cat a:profile.fix
del profile.cat
```

Figure C-2. Listing of the `DISTRIB.BAT` file to distribute courseware using multiple linked segments.

```
echo off
echo .
echo .
echo .
echo    Building installation disks for the TOPIC1A DRILL CBL
echo .
echo .
echo .
echo on

bld_inst.exe a:

cd \wordp\lesson1\topic1a\drill
del \wordp\lesson1\topic1a\drill\install\*
del \wordp\lesson1\topic1a\drill\install\*.seg
del \wordp\lesson1\topic1a\drill\install\*.pic
del \wordp\lesson1\topic1a\drill\install\*.cut
del \wordp\lesson1\topic1a\drill\install\*.pro
del \wordp\lesson1\topic1a\drill\install\*.bat
del \wordp\lesson1\topic1a\drill\install\*.exe
rmdir \wordp\lesson1\topic1a\drill\install

more < bld-errs
```

Figure C-2 (Continued)

DISTRIBUTION LIST

Assistant for Planning and Technical Development (OP-01B2)
Defense Technical Information Center (2)
Commandant, Defense Systems Management College, Ft. Belvoir, VA
Assistant Secretary of Defense (Force Management and Personnel)